
**Algoritmos bioinspirados para otimização de
CNNs aplicados na detecção de câncer de
mama utilizando imagens termográficas**

Caroline Barcelos Gonçalves



UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Uberlândia
2022

Caroline Barcelos Gonçalves

**Algoritmos bioinspirados para otimização de
CNNs aplicados na detecção de câncer de
mama utilizando imagens termográficas**

Dissertação de mestrado apresentada ao Programa de Pós-graduação da Faculdade de Computação da Universidade Federal de Uberlândia como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

Área de concentração: Ciência da Computação

Orientador: Henrique Coelho Fernandes

Coorientador: Jefferson Rodrigo de Souza

Uberlândia

2022

Ficha Catalográfica Online do Sistema de Bibliotecas da UFU
com dados informados pelo(a) próprio(a) autor(a).

G635
2022

Gonçalves, Caroline Barcelos, 1994-
Algoritmos bioinspirados para otimização de CNNs
aplicados na detecção de câncer de mama utilizando
imagens termográficas [recurso eletrônico] / Caroline
Barcelos Gonçalves. - 2022.

Orientador: Henrique Coelho Fernandes.
Coorientador: Jefferson Rodrigo de Souza.
Dissertação (Mestrado) - Universidade Federal de
Uberlândia, Pós-graduação em Ciência da Computação.
Modo de acesso: Internet.
Disponível em: <http://doi.org/10.14393/ufu.di.2022.528>
Inclui bibliografia.

1. Computação. I. Fernandes, Henrique Coelho, 1986-,
(Orient.). II. Souza, Jefferson Rodrigo de, 1985-,
(Coorient.). III. Universidade Federal de Uberlândia.
Pós-graduação em Ciência da Computação. IV. Título.

CDU: 681.3

Bibliotecários responsáveis pela estrutura de acordo com o AACR2:
Gizele Cristine Nunes do Couto - CRB6/2091
Nelson Marcos Ferreira - CRB6/3074



UNIVERSIDADE FEDERAL DE UBERLÂNDIA
 Coordenação do Programa de Pós-Graduação em Ciência da Computação
 Av. João Naves de Ávila, nº 2121, Bloco 1A, Sala 243 - Bairro Santa Mônica, Uberlândia-MG, CEP 38400-902
 Telefone: (34) 3239-4470 - www.ppgco.facom.ufu.br - cpqfacom@ufu.br



ATA DE DEFESA - PÓS-GRADUAÇÃO

Programa de Pós-Graduação em:	Ciência da Computação				
Defesa de:	Dissertação de Mestrado 12/2022, PPGCO				
Data:	16 de agosto de 2022	Hora de início:	14:00	Hora de encerramento:	15:27
Matrícula do Discente:	12012CCP001				
Nome do Discente:	Caroline Barcelos Gonçalves				
Título do Trabalho:	Algoritmos bioinspirados para otimização de CNNs aplicados na detecção de câncer de mama utilizando imagens termográficas				
Área de concentração:	Ciência da Computação				
Linha de pesquisa:	Ciência de Dados				
Projeto de Pesquisa de vinculação:	-				

Reuniu-se, por videoconferência, a Banca Examinadora, designada pelo Colegiado do Programa de Pós-graduação em Ciência da Computação, assim composta: Professores Doutores: André Ricardo Backes - FACOM/UFU, Aura Conci - UFF, Jefferson Rodrigo de Souza - FACOM/UFU (coorientador) e Henrique Coelho Fernandes - FACOM/UFU, orientador da candidata.

Os examinadores participaram desde as seguintes localidades: Aura Conci - Niteroi/RJ; André Ricardo Backes, Jefferson Rodrigo de Souza e Henrique Coelho Fernandes - Uberlândia/MG. O discente participou da cidade de Uberlândia/MG.

Iniciando os trabalhos o presidente da mesa, Prof. Dr. Henrique Coelho Fernandes, apresentou a Comissão Examinadora e a candidata, agradeceu a presença do público, e concedeu à Discente a palavra para a exposição do seu trabalho. A duração da apresentação da Discente e o tempo de arguição e resposta foram conforme as normas do Programa.

A seguir o senhor presidente concedeu a palavra, pela ordem sucessivamente, aos examinadores, que passaram a arguir a candidata. Ultimada a arguição, que se desenvolveu dentro dos termos regimentais, a Banca, em sessão secreta, atribuiu o resultado final, considerando a candidata:

Aprovada

Esta defesa faz parte dos requisitos necessários à obtenção do título de Mestre.

O competente diploma será expedido após cumprimento dos demais requisitos, conforme as normas do Programa, a legislação pertinente e a regulamentação interna da UFU.

Nada mais havendo a tratar foram encerrados os trabalhos. Foi lavrada a presente ata que após lida e achada conforme foi assinada pela Banca Examinadora.



Documento assinado eletronicamente por **Henrique Coelho Fernandes, Professor(a) do Magistério Superior**, em 16/08/2022, às 16:11, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **André Ricardo Backes, Professor(a) do Magistério Superior**, em 16/08/2022, às 16:13, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Jefferson Rodrigo de Souza, Professor(a) do Magistério Superior**, em 16/08/2022, às 16:45, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Aura Conci, Usuário Externo**, em 19/08/2022, às 05:47, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site https://www.sei.ufu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **3836984** e o código CRC **EB4EABF4**.

Este trabalho é dedicado aos meus pais João e Angelina, a minha irmã Karinne pelo apoio, incentivo, força e carinho.

Agradecimentos

Agradeço a Deus pelo dom da vida, pela presença Dele na minha vida, pela sua graça e força para que esse trabalho fosse realizado. Obrigada Senhor por ser a minha esperança e fortaleza, o meu sustento e refúgio. Obrigada Senhor porque “até aqui nós ajudou o Senhor” 1Sm 7, 12. Agradeço também a Virgem Maria, mãe de Deus e minha, pela sua intercessão e presença constante na minha vida.

Agradeço aos meus pais, João e Angelina, e minha irmã Karinne que foram essenciais em mais esse desafio. Obrigada pela presença de vocês na minha vida, pelo acolhimento de sempre, pelo incentivo, cuidado e pela paciência. Agradeço ainda a minha família, especialmente meu avô Adão que sempre me incentiva, torce e se alegra muito com meu mestrado. À minha avó, minhas tias e tios, primos e primas obrigada por estarem comigo mais uma vez ao longo desse tempo.

Agradeço aos meus orientadores, os professores Henrique e Jefferson por me acompanharem nessa jornada. Muito obrigada por cada ensinamento, pelos incentivos, pelo conhecimento compartilhado, pela paciência, por cada reunião e pelo trabalho que construímos juntos. Deus seja louvado pela vida e missão de vocês.

Agradeço aos professores da pós graduação por todo conhecimento compartilhado e ensinamentos, por abrirem meus olhos para novas áreas e possibilidades, que inclusive acabaram inspirando o tema da minha dissertação.

Agradeço aos meus amigos que estiveram comigo ao longo dessa jornada, especialmente Thaís, Filipe, Gustavo, Luan, Layse, Dry, Carla, Ana Laura, Lorraine. Obrigada pelas conversas, pelas palavras de incentivo e força, pela torcida, por compartilharmos as vitórias e desafios ao longo de mais essa etapa da minha vida.

Agradeço ainda aos meus gestores e líderes técnicos que com muita solicitude, paciência e compressão permitiram flexibilidade para que eu pudesse cumprir as responsabilidades do mestrado, participando das aulas e reuniões.

“Dele, por ele e para ele são todas as coisas. A Ele a glória por toda a eternidade!

Amém”

(Rm 11, 36)

Resumo

O câncer de mama é o segundo tipo de câncer que mais acomete mulheres no Brasil. A detecção precoce é essencial para aumentar as chances de cura e recuperação das pacientes. A termografia é uma técnica de imageamento que tem sido estudada e tem apresentado resultados promissores para auxiliar nesta detecção precoce do câncer de mama. Neste trabalho utilizamos três redes neurais convolucionais (CNNs) pré-treinadas, VGG-16, ResNet-50 e DenseNet-201, para classificar imagens termográficas estáticas da mama (em doentes ou saudáveis). Encontrar uma arquitetura para CNNs com os hiperparâmetros adequados é essencial para o bom desempenho da rede. Esta não é uma tarefa trivial e para abordar este problema utilizamos dois algoritmos de otimização bioinspirados, algoritmos genéticos (AG) e otimização por enxame de partícula (PSO), para encontrar uma boa arquitetura e bons hiperparâmetros para a parte densamente conectada das três CNNs utilizadas. Com a otimização foi possível melhorar os resultados de 0,66 para 0,92 de F1-score com o AG para a VGG e 0,83 para 0,90 de F1-score com o AG e o PSO para a ResNet, além de manter os bons resultados de 0,92 de F1-score para a DenseNet. Finalmente, buscando otimizar o tempo gasto com a execução dos experimentos do AG e do PSO, propomos a utilização de um modelo *surrogate* com *random forest*. Com o modelo *surrogate* foi possível melhorar os resultados com a DenseNet, que obteve F1-score de 1.

Palavras-chave: Termografia, Câncer de mama, CNNs, AG, PSO, *surrogate*, *random forest*.

Abstract

Breast cancer is the second most common type of cancer that affects woman in Brazil. Early detection of this disease is essential for improving chances of cure and recovery of the patients. Infrared thermography is an imaging technique being studied that has shown good potential in the early detection of breast cancer. In this work, we use three pre-trained convolutional neural networks (CNNs) VGG-16, ResNet-50 e DenseNet-201 to classify breast static thermography images (sick or healthy). Finding suitable CNN architecture and hiper-parameters is essential for a good classification outcome. Furthermore, this is not an easy task. In order to address this problem of finding suitable architecture and hiper-parameters for the fully connected part of used CNNs, we used two bio inspired optimization techniques: genetic algorithm (GA) and particle swam optimization (PSO). Using the optimization techniques we were able to improve the F1-score from 0.66 to 0.92 for the VGG using the GA and the F1-score of the ResNet from 0.83 to 0.90 using the GA and PSO. Moreover, the optimization also maintained the good results of 0.92 F1-score for the DenseNet. Finally, we opt for using a random forest surrogate model in order to obtain a cheaper model. Using the surrogate, we improved the DenseNet from 0.92 to 1 of F1-score.

Keywords: Thermography, Breast cancer, CNNs, AG, PSO, surrogate, random forest.

Lista de ilustrações

Figura 1 – Mamografias e tipos de densidade do tecido da mama. Imagens adaptadas de (SOCIETY, 2022b)	22
Figura 2 – Espectro eletromagnético - adaptado de (IBARRA-CASTANEDO, 2005)	28
Figura 3 – Captura dos dados da base DMR - extraído de (SILVA et al., 2014) . .	28
Figura 4 – Exemplo estrutura de uma CNN. Adaptado de (BALDOMINOS; SAEZ; ISASI, 2018)	29
Figura 5 – Exemplo funcionamento transferência de aprendizado. Adaptado de (TORRES-GALVÁN et al., 2021)	30
Figura 6 – Fluxograma do funcionamento de um AG adaptado de (NÉIA et al., 2013)	31
Figura 7 – Fluxograma do funcionamento de um PSO adaptado de (SHI et al., 2001)	32
Figura 8 – Fluxograma do funcionamento da <i>random forest</i> adaptado de (GUO et al., 2019)	34
Figura 9 – Visão geral da metodologia	45
Figura 10 – Exemplo paciente saudável	46
Figura 11 – Exemplo paciente doente	47
Figura 12 – Exemplo de imagens com aumento de dados	48
Figura 13 – Detalhes das redes VGG-16, ResNet-50 e DenseNet-201, adaptados de (SIMONYAN; ZISSERMAN, 2014), (HE et al., 2016), (HUANG et al., 2017).	50
Figura 14 – Exemplo da estrutura do indivíduo do AG	52
Figura 15 – Exemplo do crossover de ponto único	55
Figura 16 – Exemplo da mutação	56
Figura 17 – Definição da partícula do PSO.	58
Figura 18 – Comparação entre uma partícula do PSO e um indivíduo do AG que representam uma mesma CNN.	58
Figura 19 – Cálculo da diferença de duas partículas	59

Figura 20 – Exemplo cálculo da velocidade de uma partícula.	60
Figura 21 – Exemplo atualização da posição de uma partícula.	60
Figura 22 – Exemplo conversão de uma partícula do PSO para tamanho fixo.	62
Figura 23 – Exemplo conversão indivíduo para a entrada da <i>random forest</i>	62
Figura 24 – Curva ROC melhor resultado da DenseNet - utilizando <i>surrogate</i> e AG.	86
Figura 25 – Curva ROC melhor resultado da ResNet - utilizando <i>surrogate</i> e PSO.	86
Figura 26 – Curva ROC melhor resultado da VGG - com AG e sem o <i>surrogate</i>	87

Lista de tabelas

Tabela 1 – Comparativo de trabalhos correlatos	39
Tabela 2 – Tipo do experimento e a arquitetura correspondente para a camada densamente conectada da CNN.	64
Tabela 3 – Resultados ‘manuais’ obtidos com o grupo-1 - todas as poses.	66
Tabela 4 – Resultados ‘manuais’ obtidos com o grupo-2 - LR 0,001.	67
Tabela 5 – Resultados ‘manuais’ obtidos com o grupo-2 - LR 0,0001.	67
Tabela 6 – Resultados ‘manuais’ obtidos com o grupo-3 - LR 0,001.	68
Tabela 7 – Resultados ‘manuais’ obtidos com o grupo-3 - LR 0,0001.	69
Tabela 8 – Resultados ‘manuais’ obtidos com o grupo-4.	70
Tabela 9 – Resumo grupo-1.	70
Tabela 10 – Resumo grupo-2.	71
Tabela 11 – Resumo grupo-3.	71
Tabela 12 – Resumo grupo-4.	71
Tabela 13 – Melhores resultados com AG - grupo-2.	73
Tabela 14 – Melhores resultados com AG - grupo-3.	73
Tabela 15 – Melhores resultados com AG - grupo-4.	74
Tabela 16 – Melhores resultados com PSO - grupo-2.	76
Tabela 17 – Melhores resultados com PSO - grupo-3.	76
Tabela 18 – Melhores resultados com PSO - grupo-4.	77
Tabela 19 – Melhores resultados com <i>Surrogate</i> e AG - grupo-2.	79
Tabela 20 – Melhores resultados com <i>Surrogate</i> e AG - grupo-3.	79
Tabela 21 – Melhores resultados com <i>Surrogate</i> e AG - grupo-4.	80
Tabela 22 – Comparação AG com e sem <i>surrogate</i> - grupo-2.	81
Tabela 23 – Comparação AG com e sem <i>surrogate</i> - grupo-3.	81
Tabela 24 – Comparação AG com e sem <i>surrogate</i> - grupo-4.	81
Tabela 25 – Melhores resultados com <i>Surrogate</i> e PSO - grupo-2.	83
Tabela 26 – Melhores resultados com <i>Surrogate</i> e PSO - grupo-3.	83
Tabela 27 – Melhores resultados com <i>Surrogate</i> e PSO - grupo-4.	83

Tabela 28 – Comparação PSO com e sem <i>surrogate</i> - grupo-2.	84
Tabela 29 – Comparação PSO com e sem <i>surrogate</i> - grupo-3.	85
Tabela 30 – Comparação PSO com e sem <i>surrogate</i> - grupo-4.	85
Tabela 31 – Grupo-1 - Resultados todos os experimentos manuais LR 0,0001.	102
Tabela 32 – Grupo-1 - Resultados todos os experimentos manuais LR 0,001.	103
Tabela 33 – Grupo-2 - Resultados todos os experimentos manuais LR 0,0001.	104
Tabela 34 – Grupo-2 - Resultados todos os experimentos manuais LR 0,001.	105
Tabela 35 – Grupo-3 - Resultados todos os experimentos manuais LR 0,0001.	106
Tabela 36 – Grupo-3 - Resultados todos os experimentos manuais LR 0,001.	107
Tabela 37 – Grupo-4 - Resultados todos os experimentos manuais LR 0,0001.	108
Tabela 38 – Grupo-4 - Resultados todos os experimentos manuais LR 0,001.	109
Tabela 39 – Grupo-2 - Resultados detalhados experimentos AG	110
Tabela 40 – Grupo-3 - Resultados detalhados experimentos AG	111
Tabela 41 – Grupo-4 - Resultados detalhados experimentos AG	112
Tabela 42 – Grupo-2 - Resultados detalhados experimentos PSO	113
Tabela 43 – Grupo-3 - Resultados detalhados experimentos PSO	114
Tabela 44 – Grupo-4 - Resultados detalhados experimentos PSO	115
Tabela 45 – Grupo-2 - Resultados detalhados experimentos com surrogate e AG.	116
Tabela 46 – Grupo-3 - Resultados detalhados experimentos com surrogate e AG.	117
Tabela 47 – Grupo-4 - Resultados detalhados experimentos com surrogate e AG.	118
Tabela 48 – Grupo-2 - Resultados detalhados experimentos com surrogate e PSO.	119
Tabela 49 – Grupo-3 - Resultados detalhados experimentos com surrogate e PSO.	120
Tabela 50 – Grupo-4 - Resultados detalhados experimentos com surrogate e PSO.	121

Lista de siglas

AG Algoritmo genético

ACC Acurácia

CNN Redes neurais convolucionais, do inglês *convolutional neural networks*

CAD diagnóstico auxiliado por computador, do inglês *computer assisted diagnosis*

DMR *Database for mastology research*

GPU unidades de processamento gráfico, do inglês *graphic processing units*

INCA Instituto Nacional do Câncer

PSO otimização por enxame de partícula, do inglês *particle swarm optimization*

ROI Região de interesse, do inglês *region of interest*

RBF *Radial basis function*

RF *Random forest*

ROC *Receiver operating characteristic*

SVM *Support vector machine*

Sumário

1	INTRODUÇÃO	21
1.1	Motivação	23
1.2	Objetivos e Desafios da Pesquisa	24
1.3	Hipótese	24
1.4	Contribuições	24
1.5	Organização da Dissertação	25
2	FUNDAMENTAÇÃO TEÓRICA	27
2.1	Imagens termográficas	27
2.2	Redes neurais convolucionais	27
2.3	Algoritmos genéticos	30
2.4	Algoritmos de enxame de partícula	31
2.5	Modelos <i>surrogate</i>	32
2.6	<i>Random forest</i>	33
3	TRABALHOS CORRELATOS	35
3.1	Termografia para detecção de câncer de mama e CNNs	35
3.2	Otimização de CNNs com técnicas evolutivas	40
3.3	Modelos <i>surrogate</i>	42
4	MATERIAIS E MÉTODOS	45
4.1	Base de dados	45
4.1.1	Pré-processamento	47
4.1.2	Aumento de dados	48
4.2	Redes convolucionais	49
4.2.1	Inicializador Xavier	49
4.3	Otimização utilizando computação evolutiva	50
4.3.1	AG	51

4.3.2	PSO	56
4.4	<i>Surrogate</i>	60
5	EXPERIMENTOS E DISCUSSÕES	63
5.1	Métricas	63
5.2	Experimentos manuais	64
5.3	Experimentos com otimização	72
5.3.1	AG	72
5.3.2	PSO	75
5.3.3	Modelo <i>Surrogate</i>	78
6	CONCLUSÃO	89
6.1	Principais Contribuições	90
6.2	Trabalhos Futuros	90
6.3	Contribuições em Produção Bibliográfica	91
	REFERÊNCIAS	93

APÊNDICES 97

APÊNDICE A	–	IMAGENS UTILIZADAS	99
APÊNDICE B	–	DETALHAMENTO DOS RESULTADOS	101
B.1		Experimentos manuais	101
B.2		Experimentos com AG	101
B.3		Experimentos com PSO	101
B.4		Experimentos com surrogate	102
B.4.1		Experimentos com surrogate e AG	102
B.4.2		Experimentos com surrogate e PSO	103

Introdução

O câncer de mama é um crescimento anormal de células que começa na mama e se torna fora de controle (SOCIETY, 2022a; CâNCER, 2022). É um tipo de câncer que afeta milhares de pessoas, em sua grande maioria mulheres, em todo o mundo. De acordo com a *American Cancer Society*, o câncer de mama é o segundo tipo de câncer mais comum nos EUA, com estimativas de serem diagnosticados em 2022 cerca de 287.850 novos casos da doença (SOCIETY, 2022c). No Brasil, o Instituto Nacional do Câncer (INCA) estimou 66.280 novos casos da doença para o ano de 2021 (CâNCER, 2022). A detecção precoce do câncer de mama é essencial para que os pacientes tenham melhores chances de cura e recuperação bem como sejam submetidos a tratamentos menos invasivos (CâNCER, 2022; SOCIETY, 2022c).

Diversas técnicas de diagnóstico por imagens estão disponíveis, entre elas mamografia, ressonância magnética e ultrassonografia (CABIOĞLU; OĞUL, 2020). Mamografia é a técnica mais utilizada para auxiliar o especialista na detecção do câncer de mama. Entretanto, ela possui algumas limitações para a detecção da doença em pacientes cujas mamas são majoritariamente compostas por tecido glandular já que o tumor pode ficar oculto (CâNCER, 2022; SOCIETY, 2022b). Na Figura 1, quatro imagens de mamografias são apresentadas onde estão representados as quatro categorias de densidade da mama, a mais a esquerda é a mama majoritariamente composta por tecido adiposo e a mais a direita a mama é extremamente densa.

Termografia é uma técnica de imageamento que captura a temperatura de um corpo através de radiação infra-vermelha (ZULUAGA-GOMEZ et al., 2019; BAFFA; LATTARI, 2018). Isto é possível porque todo objeto acima do zero absoluto emite radiação infra-vermelha e essa radiação pode ser capturada através de câmeras infra-vermelhas para calcular a temperatura da superfície do objeto (ZULUAGA-GOMEZ et al., 2019). Células cancerígenas emitem mais calor que células normais devido a alta atividade metabólica dessas células (CABIOĞLU; OĞUL, 2020). Esse calor por sua vez chega até a superfície da pele o que faz com que zonas mais quentes que o normal apareçam nessa região. Além disto, a termografia é uma técnica barata e não invasiva (ZULUAGA-GOMEZ et

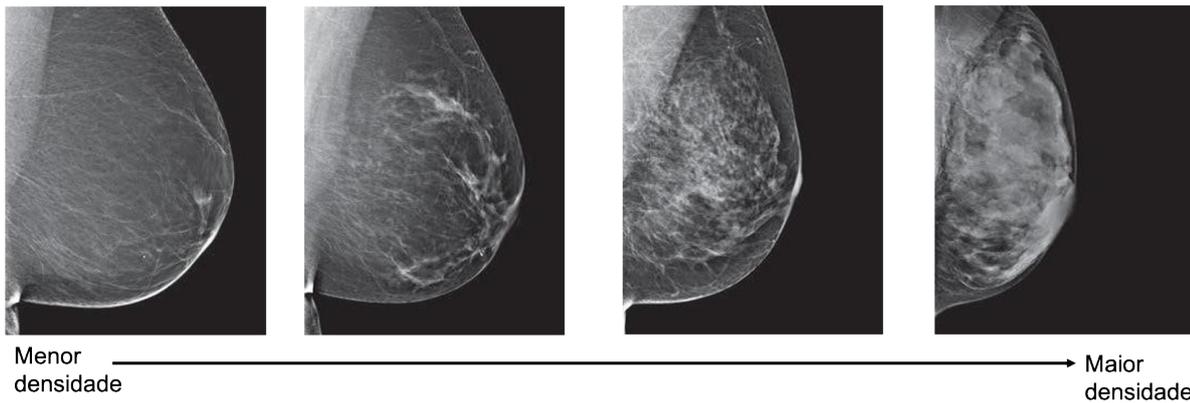


Figura 1 – Mamografias e tipos de densidade do tecido da mama. Imagens adaptadas de (SOCIETY, 2022b)

al., 2019). Por isso, as imagens termográficas podem auxiliar no diagnóstico precoce da doença principalmente para os casos onde a mamografia não apresenta bons resultados (ZULUAGA-GOMEZ et al., 2019; CABIOĞLU; OĞUL, 2020).

Redes neurais convolucionais, do inglês *convolutional neural networks* (CNN)s têm sido amplamente utilizadas para diversos problemas relacionados a visão computacional, como por exemplo classificação de imagens e reconhecimento e detecção de objetos (SUN et al., 2019b; JUNIOR; YEN, 2019; ZULUAGA-GOMEZ et al., 2019). Entre os problemas nos quais as CNNs têm sido aplicadas recentemente está a classificação de imagens termográficas (ZULUAGA-GOMEZ et al., 2019; CABIOĞLU; OĞUL, 2020; BAFFA; LATTARI, 2018).

As CNNs tem os resultados influenciados pela arquitetura e hiperparâmetros usados e portanto, desenvolver CNNs com resultados relevantes pode ser necessário o envolvimento de especialistas, demanda bastante tempo e diferentes experimentos (SUN et al., 2019a; JUNIOR; YEN, 2019). O problema de encontrar uma boa arquitetura de CNN e hiperparâmetros adequados pode ser descrito como um problema de otimização onde se busca maximizar os resultados da rede (SUN et al., 2019a). Uma alternativa para esta tarefa é a utilização de algoritmos bio-inspirados.

Algoritmo genético (AG) e otimização por enxame de partícula, do inglês *particle swarm optimization* (PSO), são dois algoritmos populacionais, meta-heurísticos e bio-inspirados que apresentam bom desempenho em problemas com amplo espaço de busca (JUNIOR; YEN, 2019; SUN et al., 2019b). Neste trabalho utilizamos esses dois algoritmos para encontrar boas arquiteturas de CNN no contexto de classificação de imagens termográficas da mama. Neste cenário, treinar um indivíduo do AG ou uma partícula do PSO envolve treinar uma CNN. Com isso, os algoritmos envolvem treinar várias CNNs.

Treinar diversas CNNs é computacionalmente caro e pode demorar várias horas e até dias (SHI; LUO; LIU, 2021). Modelos *surrogate* tem a capacidade de aproximar as funções que são originalmente bastante caras computacionalmente. Desta forma, obtendo custos

mais baixos (WANG; LIU; JIN, 2019; SUN et al., 2019a). Diferentes técnicas podem ser utilizadas como modelos *surrogate*, entre elas: processos gaussianos, *radial basis network*, otimização bayesiana, *random forest* entre outras (SUN et al., 2019a; SHI; LUO; LIU, 2021; ZHANG et al., 2021).

Este projeto de pesquisa propõe a utilização de imagens termográficas da mama, obtidas pelo protocolo estático de captura das imagens termográficas, da base pública DMR¹ para auxiliar na detecção precoce de câncer de mama através de CNNs. Além disso, propomos a utilização de algoritmos bioinspirados, AGs e PSOs, para auxiliar na otimização de hiper parâmetros de CNNs. Por fim, associamos os AGs e PSO a um modelo *surrogate* com *random forest* para possibilitar que as otimizações dos hiper parâmetros sejam menos custosas computacionalmente.

1.1 Motivação

Sistemas de diagnóstico auxiliado por computador, do inglês *computer assisted diagnosis* (CAD) têm ganhado importância porque auxiliam em diagnósticos mais precisos de forma automática e podem apoiar médicos nos diagnósticos (BAFFA; LATTARI, 2018; ZULUAGA-GOMEZ et al., 2019). Na literatura vários trabalhos tem estudado e proposto CADs para auxiliar no diagnóstico de câncer de mama (CARVALHO et al., 2022; RESMINI et al., 2021; TORRES-GALVÁN et al., 2021; ZULUAGA-GOMEZ et al., 2019).

Existem diversos trabalhos recentes envolvendo imagens termográficas da mama o que demonstra que é um tema relevante que tem sido pesquisado pela comunidade. As imagens termográficas da mama tem sido utilizadas sob as diferentes perspectivas, algumas buscando, por exemplo, melhores formas de segmentação das imagens, outros aplicando técnicas de extração de características para classificação (MAMBOU et al., 2018). Devido ao potencial que as CNNs tem apresentado para classificação de imagem, ela também tem sido utilizada em imagens termográficas da mama. Alguns trabalhos utilizam as CNNs para todo o processo (extração de características e classificação) (CHAVES et al., 2020; ZULUAGA-GOMEZ et al., 2019; CABIOĞLU; OĞUL, 2020), outros a utilizam para extração de características a serem utilizadas em outros classificadores (MAMBOU et al., 2018).

Encontrar boas arquiteturas de CNN não é uma tarefa trivial e, muitas vezes, é realizado utilizando buscas e testes manuais em um processo de tentativa e erro onde pode ser necessário apoio de especialistas e demanda muito tempo (XIAO et al., 2020; REILING et al., 2019; YOO et al., 2019). Em (YOO et al., 2019), os autores reportam um experimento que ilustra o quanto a CNN é sensível a arquitetura. Além disso, muitas vezes a quantidade de dados disponíveis para aprendizado é limitada. Para minimizar esses dois problemas, diversos trabalhos utilizam CNNs pré-treinadas, já que essas redes

¹ <http://visual.ic.uff.br/dmi/>

mostraram bons resultados em problemas e bases do estado da arte (ZULUAGA-GOMEZ et al., 2019; CABIOĞLU; OĞUL, 2020; BAFFA; LATTARI, 2018). Mesmo com redes pré-treinadas, podem ser necessários ajustes na mesma para uma melhor performance. Estes ajustes também são muitas vezes definidos através de testes manuais.

Na literatura existem trabalhos que utilizam algoritmos bioinspirados para auxiliar na escolha de boas arquiteturas de redes convolucionais, porém, esta combinação ainda não foi explorada suficientemente para imagens termográficas da mama e é justamente esse ponto que pretendemos abordar neste trabalho.

1.2 Objetivos e Desafios da Pesquisa

Este projeto de pesquisa tem como objetivo principal desenvolver uma abordagem para detecção de câncer de mama a partir de imagens termográficas estáticas utilizando CNNs e algoritmos bioinspirados para apoiarem na escolha dos parâmetros da rede.

Além disto, alguns objetivos mais específicos envolvem:

- ❑ Obter bons classificadores para imagens termográficas da mama utilizando CNNs pré-treinadas, isto é, com resultados equivalentes ou melhores aos já disponíveis na literatura.
- ❑ Propor algoritmos para encontrar bons parâmetros para a arquitetura de CNNs utilizando AGs e PSOs e desenvolver estes algoritmos para resolução de tal problema;
- ❑ Desenvolver modelos *surrogate* que possam auxiliar nos cálculos de *fitness* tanto do AG quanto no PSO.

1.3 Hipótese

Esse trabalho se baseia nas seguintes hipóteses:

- ❑ Imagens termográficas estáticas da mama podem ser utilizadas para classificar as pacientes em doentes e saudáveis com bons resultados utilizando CNNs;
- ❑ AGs e PSOs podem auxiliar na escolha de melhores parâmetros para arquitetura da CNN e com isso, melhorar os resultados da classificação;
- ❑ Modelos *surrogate* são bons otimizadores para auxiliar no cálculo das *fitness* reais.

1.4 Contribuições

As contribuições deste trabalho são:

- ❑ Comparação de três CNNs do estado da arte, VGG, ResNet e DenseNet, utilizando as imagens estáticas da base pública DMR.
- ❑ AG para otimizar redes CNNs que utilizam imagens termográficas para detecção de câncer de mama;
- ❑ PSO para otimizar redes CNNs que utilizam imagens termográficas para detecção de câncer de mama;
- ❑ Modelo *surrogate* com *random forest* utilizado em conjunto com as *fitness* reais no AG e no PSO.

1.5 Organização da Dissertação

Esta dissertação está organizada nos seguintes capítulos: O capítulo 2 aborda os principais conceitos e técnicas que serão utilizados ao longo do trabalho. No capítulo 3 alguns trabalhos correlatos são apresentados. No capítulo 4 é descrita a base utilizada, como é feito o pré processamento utilizado, detalhes das CNNs utilizadas e detalhes dos algoritmos propostos para AG, PSO e modelo *surrogate*. Os experimentos realizados e os resultados obtidos são descritos no capítulo 5. Por fim, a conclusão e contribuições deste trabalho de pesquisa está no capítulo 6.

Fundamentação Teórica

Neste capítulo serão detalhados alguns conceitos e algoritmos utilizados ao longo desta dissertação.

2.1 Imagens termográficas

Qualquer corpo que tenha temperatura acima do 0 absoluto ($0K$) emite radiação térmica (IBARRA-CASTANEDO, 2005). Esta radiação abrange ondas do espectro ultra violeta, do espectro visível e do espectro infravermelho, que são diferentes bandas do espectro eletromagnético (BORCHARTT, 2013). A Figura 2, adaptada de (IBARRA-CASTANEDO, 2005), exibe o espectro eletromagnético destacando a radiação térmica, o espectro visível e o espectro infravermelho.

A radiação infravermelha é invisível a olho nu e portanto são necessários equipamentos capazes de capturar essa radiação para ‘visualizá-la’ (IBARRA-CASTANEDO, 2005). Esses equipamentos podem ser as câmeras infravermelhas, que são capazes de capturar a radiação infravermelha e convertê-la em temperatura gerando as imagens termográficas ou termogramas (BORCHARTT, 2013). Nessas imagens, cada pixel contém um valor de temperatura associado com uma posição (um ou mais pontos) do mundo real.

A Figura 3, extraída de (SILVA et al., 2014), ilustra a captura de uma imagem termográfica da base DMR por uma câmera FLIR modelo SC620.

2.2 Redes neurais convolucionais

Redes neurais convolucionais, CNNs (do inglês *convolutional neural networks*), são redes de aprendizado profundo compostas por duas grandes etapas: a parte das camadas convolucionais e a parte densamente conectada (BALDOMINOS; SAEZ; ISASI, 2018).

A parte das camadas convolucionais é composta de uma ou mais camadas convolucionais que funcionam como extratores de características. As camadas convolucionais são responsáveis por aprender as características dos dados. Quanto mais profunda a CNN,

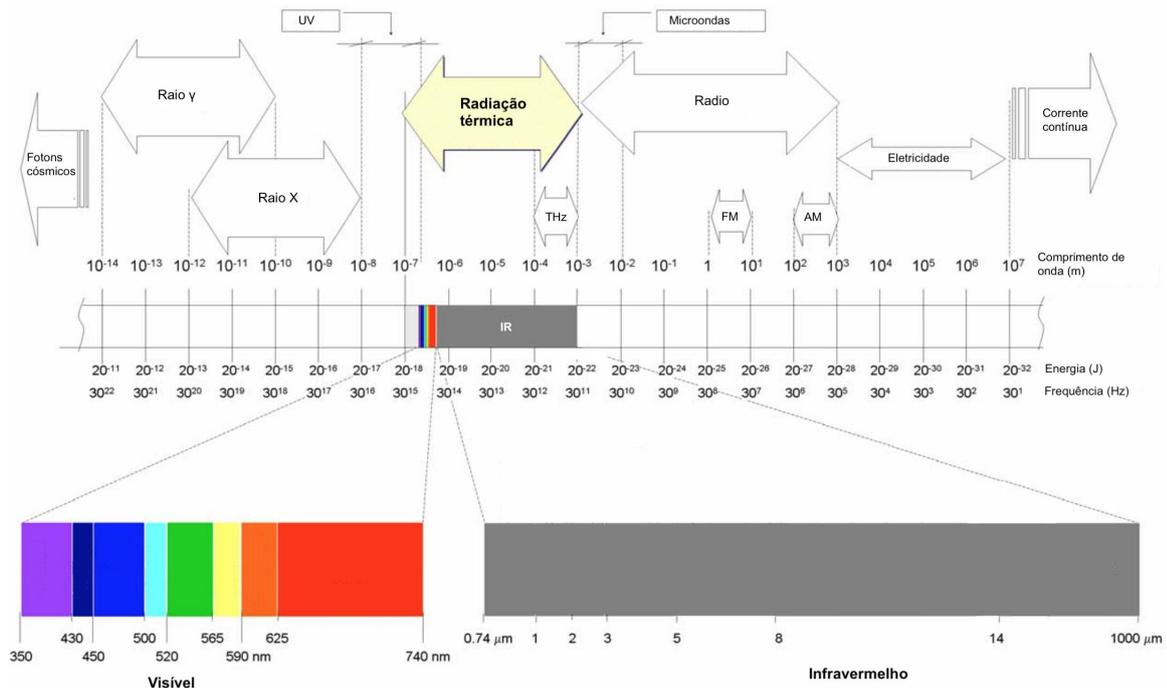


Figura 2 – Espectro eletromagnético - adaptado de (IBARRA-CASTANEDO, 2005)

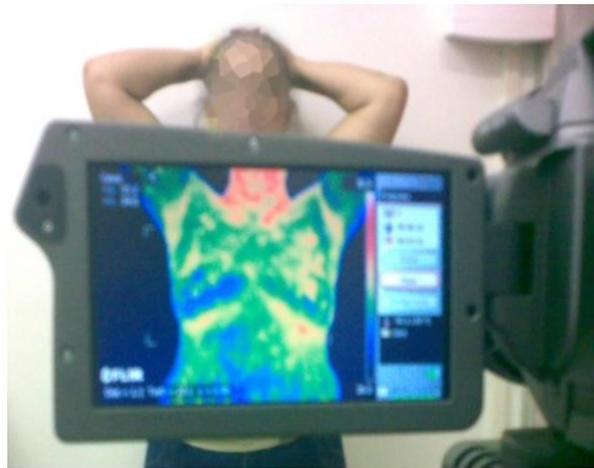


Figura 3 – Captura dos dados da base DMR - extraído de (SILVA et al., 2014)

isto é, mais camadas convolucionais, mais a rede é consegua extrair características de alto nível (BALDOMINOS; SAEZ; ISASI, 2018). Nesta parte da CNN podem também haver camadas de *pooling* que são utilizadas para reduzir a dimensionalidade dos dados e ainda para que as redes sejam invariantes a translação das imagens. Já a parte densamente conectada funciona como classificador das características extraídas na parte anterior (BALDOMINOS; SAEZ; ISASI, 2018). No geral, são utilizadas redes densamente conectadas como redes *feed-forward* ou redes recorrentes, mas em alguns casos há ainda algumas variações com adição de *dropouts* (BALDOMINOS; SAEZ; ISASI, 2018).

A Figura 4, adaptada de (BALDOMINOS; SAEZ; ISASI, 2018), ilustra uma CNN,

destacando cada parte dessa CNN.

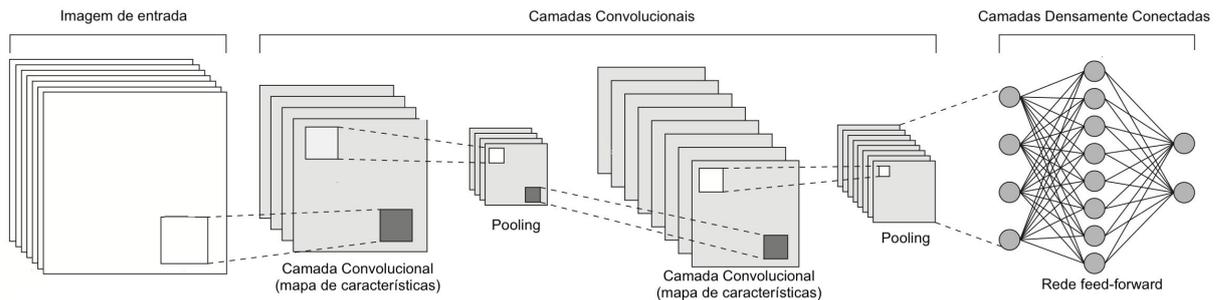


Figura 4 – Exemplo estrutura de uma CNN. Adaptado de (BALDOMINOS; SAEZ; ISASI, 2018)

As CNNs precisam de grande quantidade de dados para serem treinadas (GU et al., 2018). Porém, a disponibilidade de dados para serem utilizados pelas CNNs e a confiabilidade desses dados é um desafio (ZULUAGA-GOMEZ et al., 2019). Com isso, a técnica de aumento de dados tem sido utilizada para proporcionar a rede maior a quantidade de dados para treinamento. O aumento de dados consiste em gerar novos dados a partir de transformações dos dados disponíveis sem alterar a natureza deles. Isso pode ser feito por exemplo com transformações geométricas simples como a rotação, espelhamento, etc (GU et al., 2018). O aumento de dados também ajuda a evitar *overfitting* (CABIOĞLU; OĞUL, 2020).

Outra abordagem que também ajuda a evitar o *overfitting* é o *early stopping* (ZULUAGA-GOMEZ et al., 2019). O *early stopping* consiste em parar o treinamento uma vez que percebe-se que a rede parou de aprender (ZULUAGA-GOMEZ et al., 2019). Desta forma, as redes tem maior capacidade de generalização.

Uma técnica também bastante utilizada com as CNNs é a transferência de aprendizado. A transferência de aprendizado, ou *transfer learning*, é uma técnica onde redes pré-treinadas são utilizadas como ponto de início para novas tarefas de classificação (CABIOĞLU; OĞUL, 2020; CHAVES et al., 2020). Essas redes pré-treinadas utilizadas são treinadas em grandes quantidades de dados e possuem boa capacidade de generalização. Desta forma, o conhecimento previamente acumulado é utilizado para resolver um novo problema mais rápido e proporcionar melhores resultados (CABIOĞLU; OĞUL, 2020). Essa técnica beneficia especialmente problemas com bases de dados pequenas, onde não há quantidade de dados suficientes para bom aprendizado (TORRES-GALVÁN et al., 2021). Ao utilizar transferência de aprendizado normalmente faz-se o uso das CNNs como extractores de características de modo que apenas as camadas utilizadas para classificação são alteradas (CHAVES et al., 2020). A Figura 5 ilustra um exemplo de como a transferência de aprendizado pode ser feita. Na Figura, a CNN utilizada como exemplo é a Densenet-101. Note que a parte convolutiva, azul e laranja na Figura, é mantida inalterada em ambas as redes e apenas as etapas de classificação são alteradas.

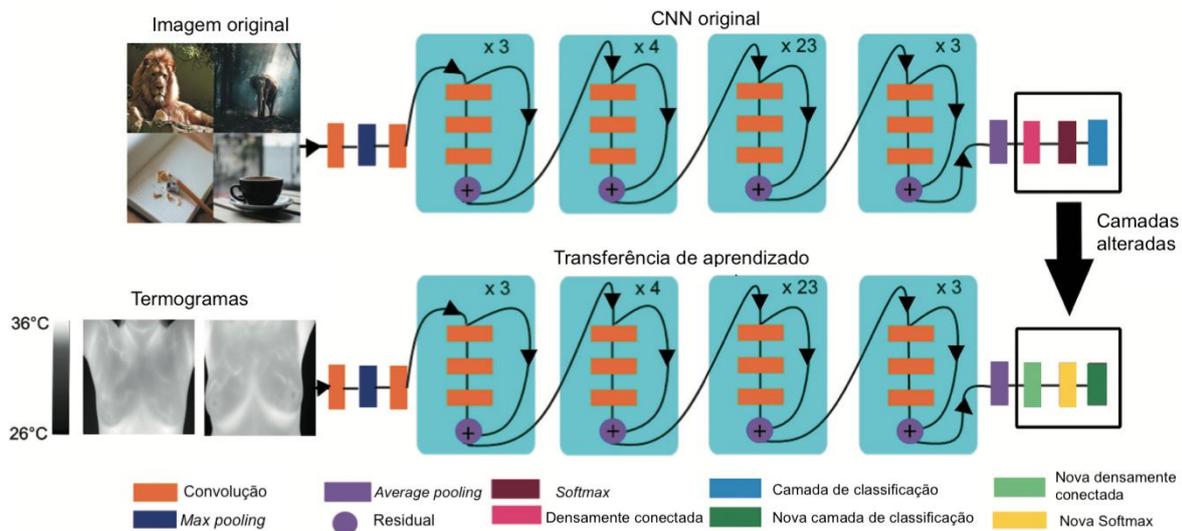


Figura 5 – Exemplo funcionamento transferência de aprendizado. Adaptado de (TORRES-GALVÁN et al., 2021)

As CNNs, mesmo as menores, têm milhares de parâmetros e portanto, dependendo da arquitetura, o treinamento pode demorar dias ou até semanas. Com isso, normalmente o treinamento de CNNs demanda a utilização de unidades de processamento gráfico, do inglês *graphic processing units* (GPU)s (JUNIOR; YEN, 2019).

2.3 Algoritmos genéticos

Algoritmo genético, AG, é um algoritmo inspirado na teoria da evolução de Charles Darwin onde os mais adaptados sobrevivem (XIAO et al., 2020). No AG, cada indivíduo é uma solução potencial, população é o conjunto desses indivíduos e genes ou cromossomos são partes desse indivíduo. Esses indivíduos são avaliados com base em funções *fitness*. Durante as gerações, que são iterações do algoritmo, indivíduos são selecionados para gerar filhos, passando pelo processo de *crossover*, e podem ainda sofrer mutações (XIAO et al., 2020). Portanto, um AG tem pelo menos três operadores bioinspirados: seleção, *crossover* e mutação.

A seleção é o operador que escolhe quais os indivíduos da população são candidatos a gerarem novos indivíduos para a próxima geração. O *crossover* é o operador que combina dois indivíduos diferentes a fim de gerar novos indivíduos. Esses indivíduos gerados herdam parte dos genes de um dos pais e o restante do outro. Por fim, a mutação é um operador que altera um ou mais genes específicos em um indivíduo (XIAO et al., 2020). A Figura 6 apresenta o fluxo básico de um AG.

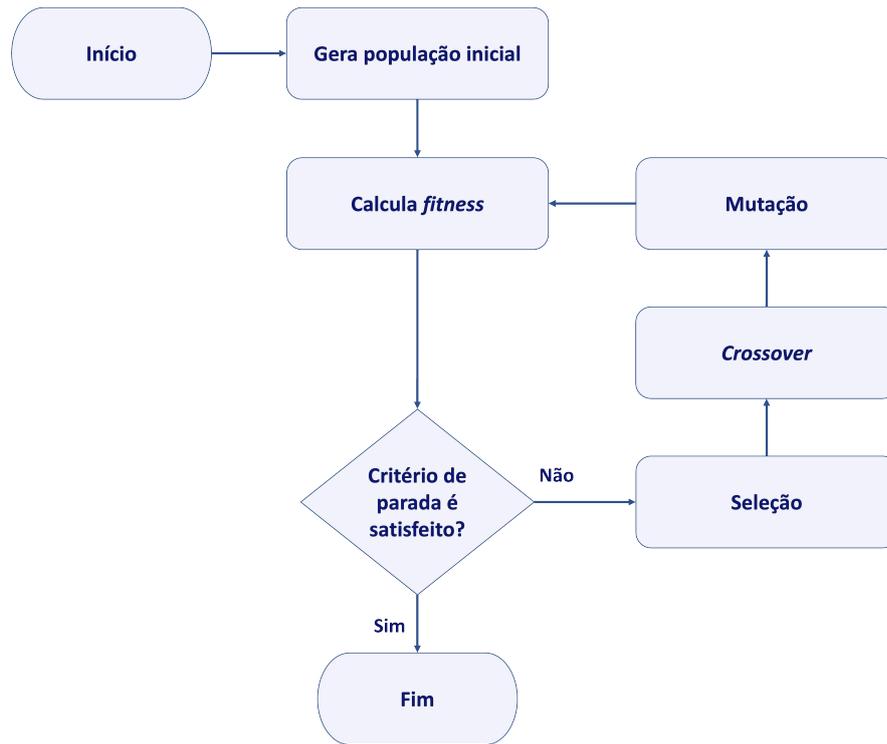


Figura 6 – Fluxograma do funcionamento de um AG adaptado de (NÉIA et al., 2013)

2.4 Algoritmos de enxame de partícula

O algoritmo de enxame de partícula, PSO (do inglês *particle swarm optimization*), é um algoritmo inspirado no padrão de voo de pássaros (SHI et al., 2001). No PSO cada possível solução potencial é chamada partícula e um conjunto de soluções potenciais é chamado enxame (SHI et al., 2001; JUNIOR; YEN, 2019).

Cada partícula tem conhecimento da sua coordenada atual no espaço de busca, da coordenada no espaço de busca que gerou a melhor posição que a partícula já esteve (chamada *pBest*) e da coordenada no espaço de busca que gerou a melhor posição entre todas as partículas (chamada *gBest*) (SHI et al., 2001; JUNIOR; YEN, 2019). A cada iteração a partícula tem sua velocidade atualizada de modo que sua posição se aproxime do melhor global e do melhor local (SHI et al., 2001; JUNIOR; YEN, 2019). A equação para atualização da velocidade é definida pela Equação 1:

$$v(t + 1) = v(t) + c_p * r_p * (pBest - x(t)) + c_g * r_g * (gBest - x(t)) \quad (1)$$

onde $v(t)$ é a velocidade da partícula no instante t , $x(t)$ é a posição da partícula no instante t , c_p e c_g são valores constantes chamados constante de aceleração (ajustando esses parâmetros, i.e. c_p e c_g , é possível proporcionar maior ou menor importância ao melhor local ou a melhor global), e r_p e r_g são valores aleatórios entre $[0, 1)$.

Após atualizar a velocidade, é preciso ainda atualizar a posição. Essa atualização é

feita pela Equação 2:

$$x(t + 1) = x(t) + v(t + 1) \quad (2)$$

A Figura 7 apresenta o fluxo básico de um PSO.

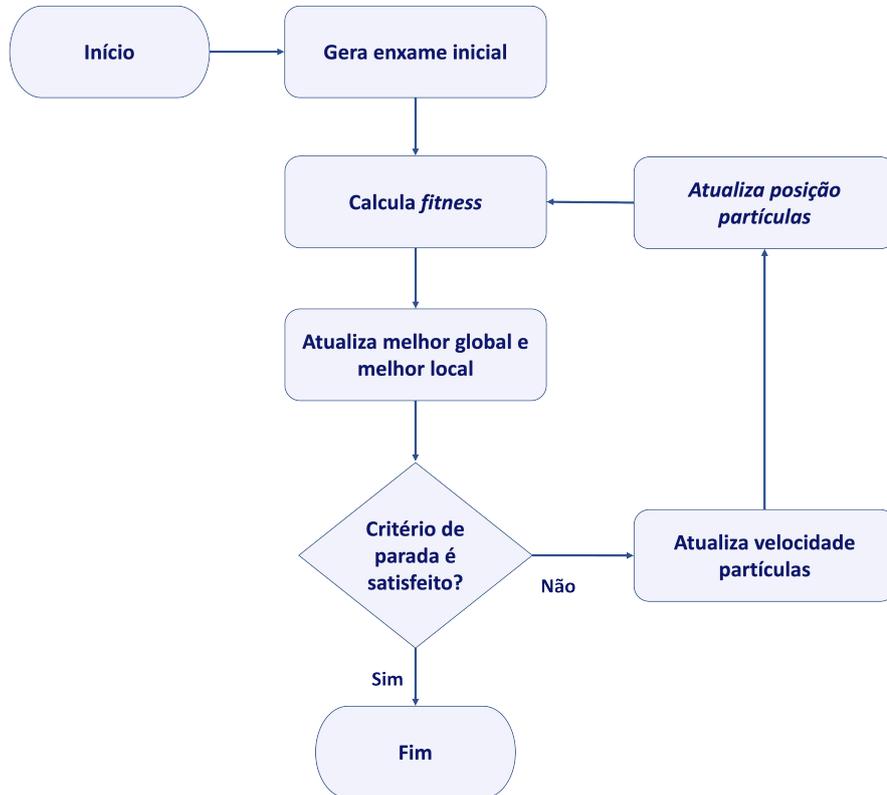


Figura 7 – Fluxograma do funcionamento de um PSO adaptado de (SHI et al., 2001)

2.5 Modelos *surrogate*

Os modelos *surrogate* são modelos de aproximação de *fitness* utilizados para apoiar ou substituir cálculos de funções *fitness* que não são triviais em algoritmos evolutivos (SUN et al., 2019a; JIN, 2011). Os *surrogates* são no geral modelos de regressão e classificação aproximados, utilizando por exemplo modelos de processo Gaussianos. Esses modelos são utilizados em conjunto com as *fitness* reais, já que são treinados com os dados disponíveis com os valores da *fitness* real (muitas vezes a quantidade de dados disponível é pequena devido a dificuldade e custo de conseguir dados reais) e então utilizados ao invés da *fitness* real (SUN et al., 2019a).

Os modelos *surrogate* podem ser divididos em modelos *online* e modelos *offline* (SUN et al., 2019a). Modelos *surrogate* são classificados como *offline* quando são treinados uma única vez com todos os dados disponíveis e depois de treinados não sofrem alterações (SUN et al., 2019a). Já os modelos *online* são modelos que podem ser atualizados a medida que

novos dados estão disponíveis. Os modelos *online* são, em alguns casos, inviáveis, quando não é possível a obtenção de novos dados. Além disso, eles podem proporcionar modelos com mais acurácia embora demandem mais tempo comparados aos modelos *offline* (SUN et al., 2019a).

Existem diferentes abordagens para a utilização dos modelos *surrogate* para o cálculo de *fitness* em algoritmos baseados em população (JIN, 2011). Cada abordagem envolve diferentes maneiras de combinar a *fitness* real com a *fitness* calculada pelo modelo durante o algoritmo evolutivo. São elas: baseada em população, baseada em geração, e baseadas em indivíduos. Na estratégia baseada em população, diferentes populações co-evoluem de forma independente e algumas dessas populações são treinadas com a *fitness* real enquanto outras são treinadas com a *fitness* calculada pelo *surrogate*. Já na estratégia baseada em geração, a divisão entre *fitness* real e *fitness* utilizando *surrogate* acontece no nível das gerações, ou seja, algumas gerações são treinadas com a *fitness* real e as outras com a *fitness* do *surrogate*. Por fim, na estratégia baseada em indivíduos, essa segregação ocorre nos indivíduos, ou seja, alguns deles tem a *fitness* calculada com o *surrogate* e outros com a *fitness* real.

2.6 *Random forest*

Random forest (RF) é uma técnica de aprendizado integrado baseado em agregação de *bootstrap* ou ensacamento (do inglês *bagging*) que utiliza diversas árvores de decisão. Essa técnica tem como vantagem poucos parâmetros para serem ajustados, não dependência de grande quantidade de dados para serem treinadas e alta robustez a ruídos e *outliers* (SUN et al., 2019a; GUO et al., 2019). A *random forest* tem aplicações tanto para tarefas de classificação quanto tarefas de regressão (SUN et al., 2019a).

No algoritmo da *random forest*, cada árvore de decisão é treinada de maneira independente entre si e utilizam uma parte dos dados de entrada (apenas um subconjunto aleatório das *features* é utilizado por cada árvore). Por fim, cada árvore faz a sua própria predição. A predição final da *random forest* é a média das predições de cada uma das árvores de decisão para casos onde a saída é contínua, ou uma votação para problemas de classificação (SUN et al., 2019a; GUO et al., 2019). A Figura 8 apresenta o fluxo básico do algoritmo da *random forest*.

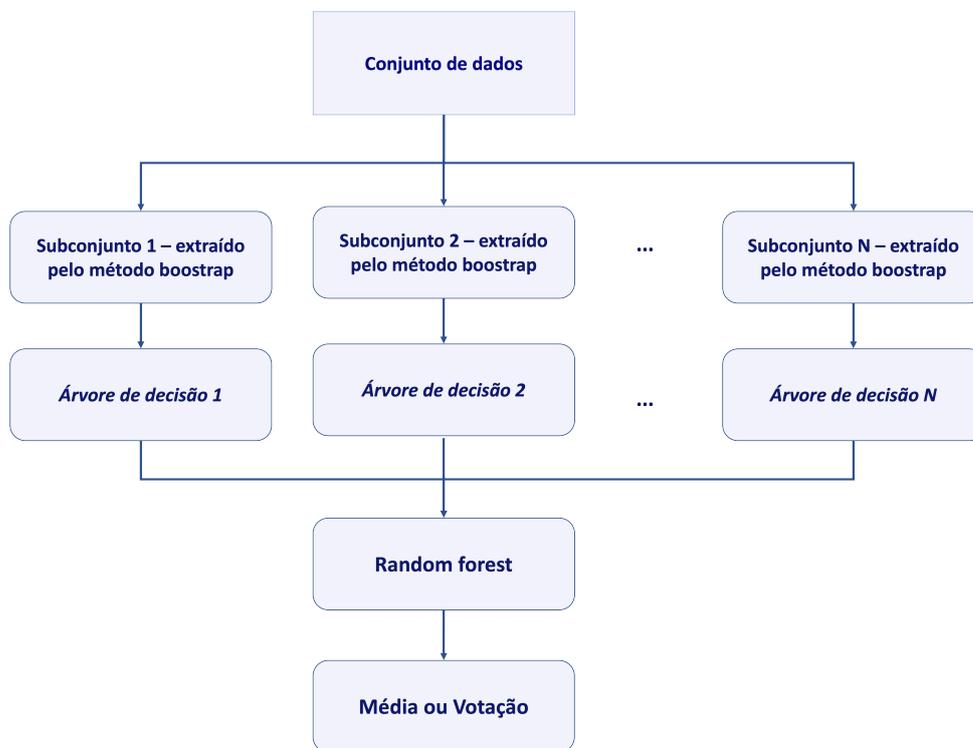


Figura 8 – Fluxograma do funcionamento da *random forest* adaptado de (GUO et al., 2019)

Trabalhos correlatos

3.1 Termografia para detecção de câncer de mama e CNNs

A base de dados *Database for mastology research* (DMR) foi apresentada em (SILVA et al., 2014). Ela é uma base de dados de imagens termográficas da mama comumente utilizada na literatura. Originalmente, a base era composta por dados de 141 pacientes coletados no Hospital Universitário Antônio Pedro (HUAP) da Universidade Federal Fluminense. Todavia, a base contou com a adição de imagens de novos pacientes ao longo do tempo e atualmente conta com imagens de aproximadamente 215 pacientes. No geral, cada paciente tem imagens coletadas tanto com protocolo estático quanto com o protocolo dinâmico. No protocolo estático uma única imagem é capturada em um dado momento, já no protocolo dinâmico, várias imagens de um paciente são capturadas durante um certo período de tempo. Cada paciente tem 5 imagens coletadas em diferentes posições utilizando o protocolo estático, sendo elas: frontal, laterais direita e esquerda 45° e laterais direita e esquerda 90°. Para o protocolo dinâmico os autores propuseram uma coleta onde ventila-se a paciente até que a média de temperatura da região entre as mamas atinja 30° ou que o tempo máximo tenha sido atingido (5 minutos). Após a ventilação, o ventilador é desligado e captura-se as imagens dinâmicas durante o retorno ao equilíbrio térmico com o meio ambiente, 20 imagens são coletadas em um período de 5 minutos e ainda 2 outras imagens laterais direita e esquerda de 90°. As imagens obtidas têm resolução de 640 x 480 pixels e cada pixel corresponde a um valor de temperatura. Os valores de temperatura variam entre 19.5° e 37°. A base, única base de dados de imagens termográficas da mama pública, está disponível publicamente em um portal disponibilizado pelos pesquisadores ¹ (SEQUEIROS, 2013).

O trabalho de (ZULUAGA-GOMEZ et al., 2019) utiliza imagens termográficas e CNNs para detecção de câncer de mama. Para isto, os autores utilizaram a base de dados de ima-

¹ <http://visual.ic.uff.br/dmi/>

gens termográficas da mama *DMR-IR*, considerando 57 pacientes do protocolo dinâmico (considerando 19 pacientes saudáveis e 37 pacientes doentes e cada paciente contendo 20 imagens). Como pré-processamento, Região de interesse, do inglês *region of interest* (ROI) foram extraídas. Diversas CNNs do estado da arte foram avaliadas, entre elas ResNet, Se-ResNet, VGG16, Inception, InceptionResNetV2 e Xception e também CNNs de propostas pelos próprios autores chamadas *baseline*. Além disso, CNNs com otimização dos hiper parâmetros foram propostas. O modelo *surrogate* proposto pelos autores tem o intuito de encontrar a melhor arquitetura de CNN para a base em questão utilizando uma otimização *bayesiana* baseada no estimador *tree parzen*. Entre os parâmetros considerados na otimização estão: número mínimo e máximo de blocos; número de camadas convolucionais e número de filtros por bloco, tipo do otimizador, tamanho *kernel*, tamanho da camada de *pooling*, tipo de *batch normalization*, taxa de *dropout*, número de neurônios conectados nas últimas duas camadas e tipo da camada top. A rede otimizada apresentou os melhores resultados quando comparada com todas as demais avaliadas, obtendo 94% de acurácia e F1-score de 0,91. Os autores afirmam que, para a base em questão, CNNs menores apresentam um melhor resultado. Outra contribuição dos autores são os experimentos relacionados a influência do aumento dos dados na classificação, buscando formas de lidar com a falta de grandes quantidades de dados disponíveis. Baseados nesses experimentos, os autores afirmam que o aumento dos dados impacta positivamente na classificação proporcionando mais informações para os modelos aprenderem. As técnicas para o aumento dos dados avaliadas pelos autores foram: inversão horizontal e vertical, rotação entre 0 e 45°, zoom (20%) e ruídos.

O trabalho apresentado em (BAFFA; LATTARI, 2018) está entre os primeiros a utilizar a base DMR-IR com CNNs. Os autores consideraram tanto as imagens do protocolo estático quanto as do protocolo dinâmico para análises e apresentaram comparativos entre os resultados obtidos com ambos os protocolos. Foram considerados para o protocolo estático 177 imagens de pacientes normais e 42 imagens de pacientes com câncer. Já para o protocolo dinâmico 95 pacientes saudáveis e 42 pacientes doentes (cada paciente com 20 imagens). Como há um desbalanceamento entre as classes, os autores propõe a utilização de corte e duplicação para aumento do número de imagens disponíveis para a classe de doentes proporcionando um melhor balanceamento das classes. A CNN utilizada foi proposta pelos próprios autores. Para utilizar as imagens dinâmicas, quatro abordagens buscando compor uma única imagem com as informações das imagens disponíveis são propostas e avaliadas. Além disso, experimentos foram realizados tanto com imagens coloridas quanto com as imagens em escala de cinza. Os autores reportam que os melhores resultados foram com imagens coloridas tanto para as imagens estáticas quanto para as imagens dinâmicas. Os resultados para as imagens estáticas obtidos foram 0,98 de Acurácia (ACC), 0,97 de sensibilidade, 1 para especificidade e F1-score de 0,98. Entre as aquisições dinâmicas consideradas pelos autores, os melhores resultados obtidos foram

0,95 de ACC, 0,93 de sensibilidade e 0,98 de especificidade e F1-score de 0,95.

O trabalho desenvolvido por (TORRES-GALVÁN et al., 2021) combina três bases de imagens infra vermelhas da mama: DMR-IR, *Instituto Jalisciense de Cancerología (IJC)* e *Instituto de Seguridad y Servicios Sociales de los Trabajadores del Estado (ISSSTE)*. Ao todo, os autores consideraram 311 imagens, das quais 267 imagens são saudáveis e 44 doentes. Utilizando a rede convolucional pré-treinada Resnet-101, os autores buscam classificar as imagens entre saudáveis e doentes. Foram realizados experimentos em dois grupos: um com balanceamento dos dados e um com a base desbalanceada, onde menos termogramas da classe doente foram utilizados. Em ambos os casos foram utilizados aumento de dados aleatórios com as seguintes técnicas: rotação entre 0° e 359° , reflexão nos eixos X e Y e translação entre 0 e 50 pixels e a combinação de todas essas técnicas. Os dados foram divididos em treinamento e teste com as proporções de 70% e 30% e apenas os dados de treinamento foram aumentados por um fator de 67 em ambos os grupos (esse fator quer dizer que uma única imagem gerou 67 novas imagens). Os resultados indicaram que utilizando os dados balanceados a rede performou melhor que com os dados desbalanceados. Para o primeiro conjunto (balanceados) os autores obtiveram 92,3% de sensibilidade contra 84,6% de sensibilidade obtido pelo segundo conjunto.

(CABIOĞLU; OĞUL, 2020) também fizeram uso das imagens estáticas da base *DMR-IR* e CNNs utilizando *transfer learning*. A CNN escolhida pelos autores foi a AlexNet. O trabalho tem por objetivo avaliar o impacto de alterar camadas específicas da rede original. Foram consideradas 181 imagens sendo 147 saudáveis e 34 doentes. Os autores também realizaram experimentos tanto com a base desbalanceada quanto balanceada. Para obter a base balanceada, aumento dos dados foi aplicado nas imagens dos pacientes doentes em alguns experimentos com as técnicas de espelhamento e rotação: -30° a 30° para obter um banco de dados balanceado. No conjunto balanceado, os autores consideraram 147 imagens saudáveis e 135 doentes. Para o pré-processamento os autores avaliaram duas formas de conversão dos valores de temperatura para o *range* que a rede espera: transformaram as imagens para o range 0-255 e replicaram cada imagem para simular 3 canais e transformaram a imagem para RGB utilizando o *colormap jet* do MATLAB (com isto as imagens ficariam coloridas). Diferentemente dos outros trabalhos, *cross validation leave one patient out* foi utilizada. Os melhores resultados obtidos foram considerando as imagens com pré-processamento RGB e aumento de dados (base balanceada) alterando-se toda a parte densamente conectada (inclusive aumentando o número de camadas) que apresentaram 94,3% de acurácia e 0,94 de F1-score.

Em (ROSLIDAR et al., 2019), os autores também utilizam a base *DMR-IR* associada a algumas redes convolucionais pré-treinadas disponíveis na literatura para detecção de câncer de mama usando a abordagem de *fine tuning*. As redes consideradas são: ResNet101, DenseNet, MobileNetV2, ShuffleNetV2. Todas elas tiveram a camada densamente conectada substituída. Foram utilizadas tanto imagens estáticas quanto imagens

dinâmicas. No treinamento apenas imagens estáticas foram utilizadas, já para teste dois grupos distintos foram considerados, um apenas com imagens estáticas e o outro apenas com imagens dinâmicas de modo que cada experimento era testado duas vezes. Além disso, foi utilizado aumento de dados no conjunto de treinamento. Os autores reportaram que a rede com melhor desempenho foi a DenseNet. Com a rede DenseNet os autores reportaram acurácia, *recall*, precisão e F1-score iguais a 1 tanto para o conjunto de imagens estáticas quanto para as imagens dinâmicas. Além disso, a MobileNetV2 apresentou bons resultados com custo computacional menor com 100% de acurácia no conjunto de teste de imagens estáticas e 99,6% de acurácia para as imagens dinâmicas.

Tanto (CHAVES et al., 2020) quanto (KIYMET et al., 2019) também utilizam CNNs juntamente com *transfer learning* na base DMR-IR. (CHAVES et al., 2020) utilizaram as imagens estáticas considerando 88 pacientes (total de 440 imagens) e as CNNs AlexNet, GoogLeNet, ResNet-18, VGG-16 e VGG-19. Os autores avaliaram diferentes *learning rates* e número de épocas. Os melhores resultados foram obtidos com a VGG-16 com 77,5% de acurácia, 85% sensibilidade e 70% especificidade. Já (KIYMET et al., 2019) consideraram 144 pacientes (88 saudáveis e 56 doentes ou suspeitos) e as seguintes CNNs: ResNet50, InceptionV3, VGG16, VGG19. Diferentes proporções de treinamento e teste foram avaliadas e ainda comparações entre o trabalho deles e trabalhos presentes na literatura que utilizam formas ‘tradicionais’ de extração de características são avaliadas. Os melhores resultados reportados foram com ResNet50 obtendo 88,89% de acurácia.

(TORRES-GALVÁN; GUEVARA; GONZÁLEZ, 2019) fazem uso da DMR-IR, considerando 173 pacientes (sendo 32 com câncer e 141 saudáveis). Os autores fazem uso do *transfer learning* considerando as seguintes CNNs pré-treinadas: AlexNet, GoogLeNet, ResNet-50, ResNet-101, Inception-v3, VGG-16, VGG-19. Para utilizar tais redes, as imagens foram redimensionadas para 224 x 224px ou 227 x 227px. Foram consideradas 70% dos dados para treinamento e 30% para teste. A rede que obteve o melhor resultado foi a VGG-16, com 100% de sensibilidade e 82,35% de especificidade.

No trabalho de (MAMBOU et al., 2018) encontra-se uma abordagem que utiliza tanto CNNs quanto *Support vector machine* (SVM) para a classificação de imagens termográficas da mama em normais ou cancerígenas. Utilizando o conceito de *transfer-learning*, a CNN utilizada é a *Inception V3* pré-treinada, que tem a camada densamente conectada alterada para classificação binária conforme o problema abordado. Caso o resultado da classificação gerado pela CNN seja uma probabilidade de incerteza da doença (entre 0,5 e 0,6), as características extraídas pela CNN servem de entrada para um SVM que faz a classificação. O SVM utilizado foi *Linear Super Vector Classification*. A base utilizada contém 67 pacientes, sendo 43 normais e 24 com a doença. Os autores reportaram bons resultados com essa abordagem, obtendo até 99% de ACC.

A Tabela 1 apresenta um comparativo entre os diferentes trabalhos que utilizam imagens termográficas apresentados nesta sessão.

Tabela 1 – Comparativo de trabalhos correlatos.

Trabalho	Banco de dados	Protocolo considerado	CNN utilizada	Melhor resultado
(ZULUAGA-GOMEZ et al., 2019)	DMR	Dinâmico	ResNet, SeResNet, VGG16, Inception, InceptionResNetV2, Xception e CNN obtida pela otimização	0,92 de F1-score com a CNN otimizada
(BAFFA; LAT-TARI, 2018)	DMR	Estático e dinâmico	CNN própria	98% de acurácia para estáticas e 95% de acurácia para dinâmicas
(TORRES-GALVÁN et al., 2021)	Combinaram três bases de dados	Estático	ResNet-101	Sensitividade de 92,6%
(CABIOĞLU; OĞUL, 2020)	DMR	Estático	AlexNet	94,3% de acurácia
(ROSLIDAR et al., 2019)	DMR	Estático e dinâmico	ResNet-101, DenseNet201 MobileNetV2 e ShuffleNetV2	100% de acurácia com a DenseNet
(CHAVES et al., 2020)	DMR	Estático	AlexNet, GoogLeNet, ResNet-18, VGG-16 e VGG-19	77,5% com VGG-16 e VGG-19
(KIYMET et al., 2019)	DMR	Não mencionado	VGG16, VGG19, ResNet-50 e InceptionV3	88,89 de acurácia com a ResNet
(TORRES-GALVÁN; GUEVARA; GONZÁLEZ, 2019)	DMR	Não mencionado	AlexNet, GoogLeNet, ResNet-50, ResNet-101, Inception-v3, VGG-16 e VGG-19	91,18% de acurácia balanceada
(MAMBOU et al., 2018)	DMR	Dinâmico	Inception V3 com SVM	ROC de 1

Comparativo de trabalhos correlatos

Diversos outros trabalhos utilizam imagens termográficas porém com outros classificadores como é o caso de (RESMINI et al., 2021). Os autores em (RESMINI et al., 2021) utilizam imagens termográficas estáticas de duas bases de dados DMR e uma base privada da Universidade Federal de Pernambuco. Os autores realizaram a extração de ROI nas imagens. A abordagem proposta envolve tanto a seleção de modelos quanto de características utilizando AG e SVM. Na primeira etapa os autores utilizam a técnica de *bucket of models* com um AG para seleção dos modelos SVM que são treinados. Nesta etapa, os autores definiram diferentes parâmetros para o SVM a serem avaliados. Ao final desta etapa, um conjunto com os melhores modelos são obtidos. Já na segunda etapa outro AG é utilizado porém com o intuito de selecionar as características que melhores descrevem os dados utilizando os modelos obtidos na primeira etapa. Após essas duas etapas, o modelo resultante é então utilizado para classificar os pacientes em saudáveis ou doentes. Os

resultados obtidos mostram que a seleção das características melhorou significativamente os resultados, sendo que os melhores resultados obtidos apresentaram 97.18% de acurácia e 97.91% de AUC. Além disso, os resultados obtidos foram equivalentes a outros trabalhos que utilizam diferentes abordagens como *deep learning*.

Em (GLOROT; BENGIO, 2010), os autores trazem um estudo sobre as funções de ativação em redes profundas e possíveis causas para a função de ativação padrão em redes neurais (sigmoide) não performar bem nessas redes profundas. O estudo avalia três funções de ativação diferentes sigmoide, tangente hiperbólica e *softsign*. Os autores perceberam que a sigmoide desacelera o aprendizado e ocasiona saturação precoce da camada oculta superior. Essa saturação impede que os gradientes façam boa propagação para a correção dos pesos (*backpropagation*). Os autores propõe, considerando funções de ativação linear (simétricas), um método chamado *normalized initialization* para inicialização do pesos da rede que visa manter as variações de ativação da rede e também a variação dos gradientes *back-propagated* à medida que se caminha na rede. Essa técnica, que ficou conhecida como *inicializador Xavier*, considera a inicialização baseada em uma distribuição uniforme que tem como parâmetro o tamanho de cada camada. Os testes indicaram que a técnica de inicialização proposta melhorou os resultados com a função de ativação tangente hiperbólica, enquanto as redes com a função *softsign* se mostraram resistentes as diferentes formas de inicialização.

3.2 Otimização de CNNs com técnicas evolutivas

O trabalho apresentado em (SUN et al., 2019b) utiliza CNN para classificação de imagens em 9 bases *benchmarks*, entre elas: *Fashion*, *Rectangle Images*, *Convex Sets*, *MNIST Basic*, *MNIST with Rotated Digits*, *MNIST with RD plus Background Images (MRDBI)*. Os autores propõe um novo método utilizando AGs para encontrar melhores arquiteturas e pesos de conexão iniciais para CNNs em problemas de classificação de imagens. Os parâmetros considerados na otimização foram: altura e largura do filtro, número de mapas de características, altura e largura do *stride*, tipo de convolução, média e desvio padrão dos elementos de filtro (genes da camada de convolução); altura e largura do *kernel*, altura e largura do *stride* e tipo de *pooling* (genes da camada de *pooling*); número de neurônios, média e desvio padrão da conexão dos pesos (genes da camada densamente conectada). No AG proposto, os indivíduos têm genes de tamanhos variáveis onde cada gene codifica a informação ou parâmetros de uma camada da CNN. Devido a essa variação no tamanho dos indivíduos, o cruzamento é feito por um *método de alinhamento de unidades* proposto pelos autores. O método de seleção adotado foi torneio binário. Foi aplicado elitismo com taxa de 20% além de mutação (podendo acontecer adicionando, excluindo ou alterando um gene). A função *fitness* utilizada é composta por três informações: média, desvio padrão (da média dos erros) e número de pesos. A CNN

depois de otimizada é então comparada com diversas redes do estado da arte entre elas: GoogleNet, AlexNet, VGG16, ScatNet-2, RandNet-2, etc. Em 6 das 9 bases, os autores reportaram um desempenho melhor que os melhores resultados anteriormente reportados. Embora para as outras 3 bases não obtiveram resultados superiores de ACC, obtiveram bons resultados com redes muito menores. Os autores realizaram ainda experimentos avaliando o impacto da inicialização. Para isto, compararam as redes ‘evoluídas’ com e sem o inicializador de Xavier. Percebeu-se que o inicializador de Xavier teve impacto positivo, de forma que as redes inicializadas com esta técnica performaram melhor para todas as bases.

O trabalho de (BAKHSI et al., 2019) também propõe CNNs cujas arquiteturas foram otimizadas por AG, porém, com foco apenas nos hiperparâmetros. Os hiperparâmetros considerados foram: número de camadas da rede, número de mapas de características, taxa de aprendizado, peso do fator de decaimento e momento. A função *fitness* é obtida pela média de ACC dos dados do conjunto de validação. Utiliza-se *crossover* uniforme com a presença de elitismo e mutação (a mutação consiste em alterar aleatoriamente dos parâmetros). As bases consideradas foram CIFAR10 e CIFAR100. Os autores comparam os resultados obtidos com a CIFAR10 com outros trabalhos reportados na literatura. Esses trabalhos foram divididos conforme as técnicas que são utilizados, nas seguintes classes: 100% manuais, semi-automáticos e automáticos. A rede proposta obteve melhores resultados para o primeiro grupo. Embora ela não apresentou resultados superiores para todos os trabalhos dos demais grupo, ela obteve bons resultados com tempo e menores arquitetura da rede que as utilizadas para comparação no estado da arte. Já a base CIFAR100 foi avaliada fazendo o uso da técnica de *transfer learning* a partir da melhor CNN obtida com a CIFAR10 e obteve bons resultados embora não os melhores em todos os grupos avaliados.

(JUNIOR; YEN, 2019) propõe o algoritmo psoCNN que utiliza otimização de enxame de partículas (PSO) para encontrar boas arquiteturas de CNNs com convergências mais rápidas que outras técnicas evolutivas. Os autores definem partículas com tamanhos dinâmicos compostas de múltiplos tipos de camadas: convolucionais, *max pooling*, *average pooling* e cada tipo de camada tem seus próprios hiper parâmetros. Além disto, métodos para o cálculo da diferença de velocidade e atualização da velocidade capazes de lidar com os tamanhos variáveis da partícula são propostos. Os autores fazem uso do inicializador Xavier para inicializar os pesos das CNNs. Para validar o algoritmo, os autores utilizaram diversas bases públicas, entre elas: MNIST, Rectangles-I, Convex, e MNIST-Fashion e compararam com diversos resultados da literatura. Os autores reportam valores de erros similares a alguns trabalhos da literatura porém com arquiteturas menores.

Em (SILVA et al., 2018), os autores também fazem uso do PSO para auxiliar na escolha de alguns parâmetros de CNNs para classificar imagens de tomografia do pulmão buscando identificar nódulos. A base utilizada é LIDC-IDRI. Os parâmetros considerados

na otimização são número de filtros em cada camada convolucional, número de neurônios na camada densamente conectada, tamanho dos filtros, tipo de *pooling*, tamanho do *batch*, probabilidade de *dropout* nas camadas convolucionais e densamente conectada. Além disso, cada um deles tem um intervalo mínimo e máximo para evitar o aparecimento de redes muito grandes e ocasionar *overfitting*. A *fitness* considerada pelos autores é composta de uma função que leva em conta acurácia, sensibilidade e especificidade. Os autores obtiveram resultados significativos a outros trabalhos da literatura, superando diversos resultados e, embora tenham obtido resultados inferiores em alguns cenários, ainda assim foram bons resultados.

3.3 Modelos *surrogate*

O trabalho desenvolvido por Sun et al. em (SUN et al., 2019a) propõe um modelo *surrogate* baseado em *random forest* que é utilizado como *fitness* em algoritmos evolutivos que são utilizados para otimizar arquiteturas de CNNs. O modelo proposto envolve um algoritmo *offline* utilizado para substituir *fitness* reais e caras que normalmente envolvem treinar CNNs do zero. O algoritmo proposto é composto de duas grandes partes: a primeira parte é onde a *random forest* é treinada com dados de CNNs previamente treinadas do zero com a respectiva acurácia; e, a segunda parte é o algoritmo evolutivo em que a *fitness* é o modelo da *random forest* treinada na etapa anterior. Na primeira etapa do modelo, as CNN treinados do zero são codificados em um indivíduo com dois tipos diferentes de variáveis: o primeiro é uma tripla com (tipo, saída e quantidade), onde tipo se refere ao tipo de bloco que pode ser Blocos da DenseNet ou Blocos da ResNet, saída refere-se ao número de neurônios de saída dessa camada e quantidade equivale ao número de blocos dessa configuração; o segundo tipo de variável é uma tupla com tipo de *pooling* e posição de camada onde o tipo do *pooling* se refere ao qual tipo de *pooling* utilizado que pode ser *pooling* máximo ou *pooling* médio, e posição de camada se refere a onde esse *pooling* deve ser colocado. Após a codificação das CNNs treinadas, a *random forest* é então treinada. O algoritmo evolucionário não treina as CNNs para calcular a *fitness*, ao invés, utiliza o modelo da *random forest* treinado anteriormente em todas as iterações para toda a população. Para validar o modelo proposto, os autores fizeram comparações com CNNs do estado da arte (VGG, por exemplo), CNNs com arquitetura escolhidas por algoritmos evolutivos e por algoritmos não evolutivos em duas bases de dados diferentes: CIFAR10 e CIFAR100. O classificador proposto foi capaz de reduzir significativamente o tempo e, ao mesmo tempo, melhorar a acurácia na maioria nas comparações feitas pelos autores.

(JIN, 2011) traz uma *survey* sobre modelos *surrogates*. Segundo ele, diversos problemas tem funções de avaliação custosas e/ou complexas computacionalmente. Desta forma, modelos *surrogates* foram propostos para auxiliar nestes cenários para reduzir o

tempo computacional. De acordo com o autor, modelos *surrogates* devem ser utilizados em conjunto com a *fitness* real e existem algumas abordagens possíveis para essa combinação entre quando executar o *surrogate* e quando executar a *fitness* real, são elas baseados: no indivíduo, na população e nas gerações. Na primeira abordagem alguns dos indivíduos são escolhidos para terem a *fitness* real calculada e para os demais o cálculo é do *surrogate*. Na abordagem de população, mais de uma população co-evolui e cada uma tem seu próprio *surrogate*. Por fim, na abordagem de geração, em algumas gerações a *fitness* real é calculada, enquanto nas demais o *surrogate* é calculado.

Em (ZHANG et al., 2021) os autores propõem um modelo *surrogate* gaussiano não estacionário e baseado em *kernel* (funções de covariância) utilizando estratégias evolutivas. Os autores chamam o modelo de *HOUSES*, que é uma abreviatura para *Hyperparameter Optimization with sUrrogate-aSsisted Evolutionary Strategy*. Embora eles não afirmem utilizar algoritmos genéticos, diversos conceitos de AGs são utilizados: presença de população, seleção, mutação. Os autores propõem ainda uma função de covariância não estacionária. Para avaliar o modelo proposto são utilizadas a base MNIST (o banco de dados MNIST é um grande banco de dados de dígitos manuscritos comumente usado para treinar vários sistemas de processamento de imagem) com algumas redes do estado da arte, como a LeNet5 e AlexNet, onde diferentes hiperparâmetros foram selecionados, entre eles número de neurônios nas camadas densamente conectadas, taxa de *dropout* entre as camadas densamente conectadas. Além dessa base, foram utilizadas duas bases de imagens de tomografia do pulmão (*Lung Image Database Consortium (LIDC)* and *Image Database Resource Initiative (IDRI)*) buscando classificar os tipos de nódulos. O modelo proposto é então comparado com diferentes outras técnicas para modelos surrogate, são elas: busca aleatória *random search*, *grid search*, *Tree-structured Parzen Estimator*, processos gaussianos com *kernel* estacionários, otimização via *Radial basis function* (RBF) e busca de coordenadas dinâmicas. Os autores afirmam que para problemas onde a dimensionalidade dos hiperparâmetros não é muito grande, algoritmos de busca aleatória simples proporcionam resultados satisfatórios e ainda que o algoritmo proposto encontrou melhores hiperparâmetros nas bases e comparações realizadas.

Materiais e métodos

Neste capítulo será apresentada a metodologia utilizada neste trabalho. Detalharemos a base de dados utilizada, as CNNs utilizadas, o AG e o PSO propostos. A Figura 9 apresenta uma visão geral da metodologia proposta nesta dissertação.

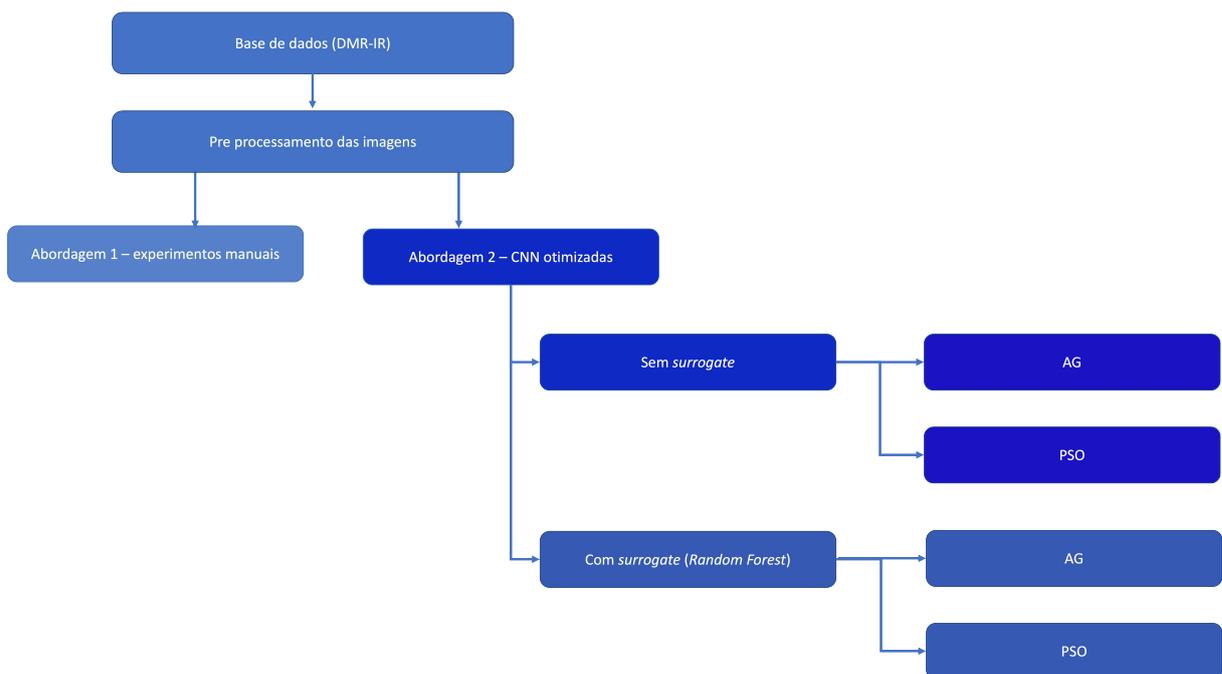


Figura 9 – Visão geral da metodologia

Todo código desenvolvido neste projeto está disponível *online* no Github.¹

4.1 Base de dados

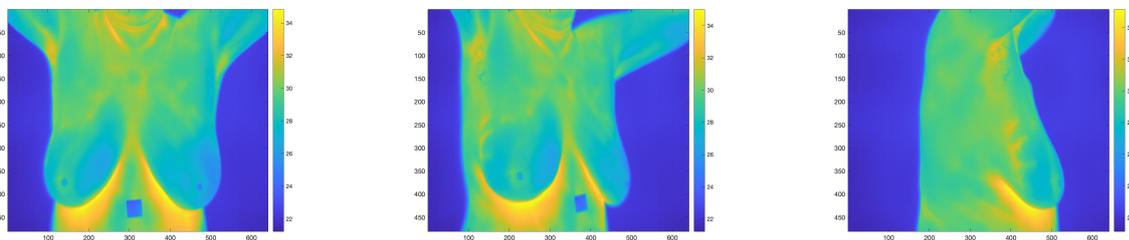
A base de dados utilizada neste trabalho é a base *DMR*. Esta base foi escolhida devido a sua relevância na literatura para trabalhos relacionados a imagens termográficas da mama,

¹ <https://github.com/Carolbg/mestrado-experimentos>

já que é a única base de dados pública e conta com um número relevante de imagens. A base se encontra disponível *online* (SEQUEIROS, 2013). As imagens da base têm resolução 640x480 px onde cada pixel tem um valor de temperatura. Os valores variam entre aproximadamente 19.5° e 37°. ²

A base conta com imagens de duas grandes classes: imagens de pacientes saudáveis e imagens de pacientes doentes (que são pacientes com câncer de mama). Há ainda algumas imagens de pacientes cuja classe é desconhecida. Todas as imagens desta última classe foram desconsideradas em todas as análises. Na base, existem imagens de dois protocolos de aquisição diferentes, imagens estáticas e imagens dinâmicas. As imagens estáticas são coletadas após um processo de aclimação da paciente, onde cada paciente tem no geral 5 imagens coletadas, cada uma em uma pose diferente, sendo que as poses são: frontal, laterais direita e esquerda com 45° e laterais direita e esquerda com 90°. Já a coleta das imagens dinâmicas também ocorrem após o processo de aclimação da paciente proposto pelos autores. Porém, 20 imagens são coletadas em uma mesma posição frontal em um intervalo de tempo de 5 minutos. Há ainda no protocolo dinâmico a coleta de outras 2 imagens tiradas em poses laterais de 90°. Ao todo, a base contém aproximadamente 5464 imagens, sendo 1057 imagens estáticas (destas imagens 864 são imagens de pacientes saudáveis e 193 imagens de pacientes doentes) e 4407 imagens dinâmicas (sendo 3580 imagens de pacientes saudáveis e 827 imagens de pacientes doentes).

Neste trabalho optamos por utilizar as imagens do protocolo estático. Na base estão disponíveis dados de 215 pacientes com imagens estáticas sendo 176 pacientes saudáveis (totalizando 864 imagens da classe saudável) e 39 pacientes doentes (totalizando 193 da classe doente). Algumas imagens da base foram descartadas por diferentes motivos como: imagem embaçada, imagem com pontos, imagens com mancha, entre outros. As Figuras 10 e 11 contém alguns exemplos das imagens utilizadas com as 5 posições diferentes disponíveis.

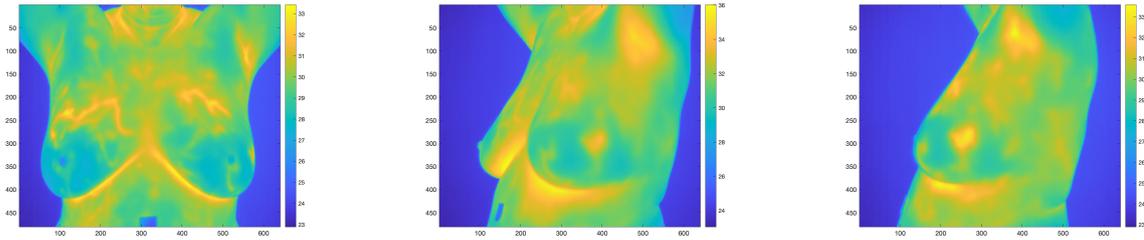


(a) Paciente saudável - frontal (b) Paciente saudável - 45° (c) Paciente saudável - 90°

Figura 10 – Exemplo paciente saudável

Embora amplamente utilizada na literatura, diferentes trabalhos utilizam diferentes número de imagens da base DMR. Além disso, não temos a informação específica de quais

² <http://visual.ic.uff.br/dmi/>



(a) Paciente doente - frontal (b) Paciente doente - 45° (c) Paciente doente - 90°

Figura 11 – Exemplo paciente doente

imagens da base foram utilizadas nos diferentes trabalhos. Com isso, foram escolhidos dois subconjuntos da base: um com 34 pacientes doentes e 147 pacientes saudáveis (esses valores foram escolhidos conforme o trabalho de (CABIOĞLU; OĞUL, 2020) para fins de comparação posteriormente); um subconjunto balanceado com 38 pacientes de cada classe.

4.1.1 Pré-processamento

As imagens originais utilizadas neste trabalho são imagens 2D cujos valores dos pixels correspondem aos valores de temperatura de cada região. Desta forma, uma etapa que se faz necessária é a conversão das imagens para o padrão de entradas que as redes pré treinadas utilizadas esperam. As redes consideradas esperam imagens com dimensão 224 x 224 pixels, cada uma com 3 canais e valores de pixels no intervalo [0,1]. Primeiramente as imagens foram, portanto, redimensionadas. Em seguida, para a ‘conversão’ em três canais utilizamos duas abordagens. A primeira consiste em utilizar o min-max para ajustar o intervalo dos valores e aplicar a replicação dos canais. A fórmula do min-max é apresentada pela Equação 3.

$$\bar{X} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (3)$$

onde \bar{X} é o valor resultante do pixel, X é o valor original do pixel (temperatura), X_{max} e X_{min} são os valores mínimos e máximos da imagem. A outra etapa é a replicação do canal, onde o canal obtido é replicado outras duas vezes (gerando então três canais iguais).

A segunda abordagem é chamada de abordagem com JetColormap. Ela é baseada no trabalho de (CABIOĞLU; OĞUL, 2020) que utiliza-se do visualizador de imagens do MATLAB. Ao visualizar as imagens nesse software com o *Jet* como *colormap* é gerada uma visualização da imagem com 3 canais com valores diferentes para cada canal. Nessa abordagem, buscamos pré processar as imagens para obter resultados similares a visualização disponível. Essa abordagem envolve a conversão para uma *index image* do MATLAB e em seguida a conversão dessa *index image* para uma imagem em RGB. A

utilização desta forma de pré-processamento busca prover para as redes imagens mais similares as utilizadas originalmente na etapa de treinamento.

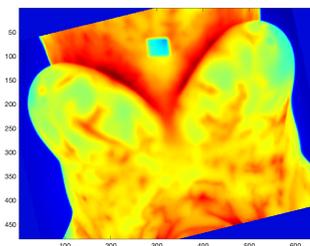
Finalmente, a última etapa de pré-processamento, que é aplicada para ambas abordagens, é a normalização da base. Isso foi realizado utilizando a média e desvio padrão da base utilizada de todas as imagens consideradas de acordo com a Equação 4:

$$\text{Imagem normalizada} = \frac{\text{imagem} - \text{média}}{\text{desvio padrão}} \quad (4)$$

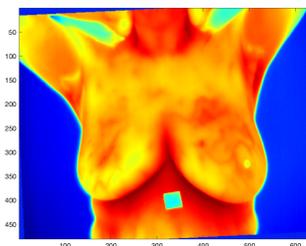
onde imagem é a imagem a ser normalizada, *média* e desvio padrão são respectivamente a média dos pixels de todas as imagens utilizadas (subconjunto de imagens da DMR utilizado) e o desvio padrão das imagens utilizadas.

4.1.2 Aumento de dados

As técnicas de aumento de dados tem sido amplamente utilizadas na literatura por propiciar a utilização de bases desbalanceadas e/ou com poucos dados (CABIOĞLU; OĞUL, 2020; BAFFA; LATTARI, 2018). Desta forma, optamos por também considerar o aumento de dados em alguns dos experimentos realizados neste trabalho. As técnicas de aumento de dados consideradas foram espelhamentos horizontais e verticais bem como rotação com um ângulo aleatório escolhido entre -30° e 30° . Todos esses aumentos de dados são aplicados de forma aleatória, portanto uma imagem pode ter os três tipos, enquanto outra pode ter apenas um. A Figura 12 contém uma imagem que ilustra duas das imagens após o aumento dos dados. As figuras demonstradas fazem uso do pré-processamento do *Jet Colormap* mencionando anteriormente e por isso tem cores bem características desse mapa de cores do MATLAB. Note que embora ambas as figuras tenham rotação aplicada, cada uma tem um ângulo diferente de rotação e, além disso, uma tem espelhamento e a outra não. Isso ocorre devido a aleatoriedade com que o aumento de dados é realizado.



(a) Imagem de paciente saudável após espelhamento e rotação de x° .



((b) Imagem de paciente doente após rotação de y° .

Figura 12 – Exemplo de imagens com aumento de dados

4.2 Redes convolucionais

Diversas redes convolucionais pré-treinadas estão disponíveis na literatura, redes essas que foram treinadas com a base Imagenet (uma base contém milhares de registros). Essas redes tem bons extratores de características que poderiam ser reutilizados para novos problemas. Desta forma, selecionou-se três redes com base em dois critérios: i) comuns entre os trabalhos que utilizam imagens termográficas e ii) mostraram bons resultados para o problema em questão. Com base nestes critérios, foram selecionadas as seguintes CNNs: VGG-16 (proposta por (SIMONYAN; ZISSERMAN, 2014)), ResNet-50 (proposta em (HE et al., 2016)) e DenseNet-201 (proposta em (HUANG et al., 2017)). A Figura 13 contém os detalhes da arquitetura de cada uma dessas redes.

Todas as três redes foram utilizadas fazendo uso da transferência de aprendizado, alterando apenas a parte da camada densamente conectada e mantendo as camadas convolucionais que atuam como extratores de características das redes originais. Na Figura 13 as partes alteradas são os posteriores aos últimos *poolings*, i.e., todas as camadas que começam com ‘FC-’ e o *softmax*.

Também se faz necessário definir alguns parâmetros destas redes, por exemplo, taxa de aprendizado a ser utilizada, quantas camadas serão utilizadas na parte densamente conectada da rede e quantos neurônios devem estar presentes em cada camada e ainda se deve ou não ser utilizada *dropout* nessas camadas. Para a escolha destes parâmetros foram utilizadas inicialmente escolhas e testes manuais descritos na seção (5.2). A posteriori, estes parâmetros foram escolhidos através de algoritmos evolutivos de otimização (AG e PSO) descritos na próxima seção.

4.2.1 Inicializador Xavier

No trabalho de (GLOROT; BENGIO, 2010), os autores propõe um método para inicialização de pesos de redes profundas que proporciona convergência mais rápida das mesmas. O método, que ficou conhecido como ‘inicializador de Glorot’ ou ‘inicializador Xavier’, faz uso de uma distribuição uniforme para a inicialização dos pesos que é definida conforme a Equação 5:

$$U \left[-\frac{\sqrt{6}}{\sqrt{n_j + n_{j+1}}}, \frac{\sqrt{6}}{\sqrt{n_j + n_{j+1}}} \right] \quad (5)$$

onde n é o tamanho da camada.

Alguns trabalhos, como (JUNIOR; YEN, 2019) e (SIMONYAN; ZISSERMAN, 2014), utilizam ou referenciam o inicializador Xavier. Desta forma, optamos por também o utilizar. O inicializador Xavier é utilizado para inicializar os pesos das camadas densamente conectadas em todos os experimentos realizados. Como os pesos das camadas convolucio-

	VGG-16	ResNet-50	DenseNet-201
	Imagem input	Imagem input	Imagem input
	Conv3-64 Conv3-64	conv1 7x7, 64, stride 2	Convolution 7x7 conv, stride 2
	maxpool	3x3 max pool, stride 2	Pooling 3x3 max pool, stride 2
	Conv3-128 Conv3-128	conv2_x $\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	Dense Block (1) $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
	maxpool		Transition Layer (1) 1 x 1 conv 2 x 2 average pool, stride 2
	Conv3-256 Conv3-256 Conv3-256	conv3_x $\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	Dense Block (2) $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$
	maxpool	conv4_x $\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	Transition Layer (2) 1 x 1 conv 2 x 2 average pool, stride 2
	Conv3-512 Conv3-512 Conv3-512	conv5_x $\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	Dense Block (3) $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$
	maxpool		Transition Layer (3) 1 x 1 conv 2 x 2 average pool, stride 2
	Conv3-512 Conv3-512 Conv3-512		Dense Block (4) $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$
Camadas de classificação	maxpool FC-4096 FC-4096 FC-1000 soft-max	Average pool FC-1000 soft-max	7x7 global pool FC-1000 softmax

Figura 13 – Detalhes das redes VGG-16, ResNet-50 e DenseNet-201, adaptados de (SIMONYAN; ZISSERMAN, 2014), (HE et al., 2016), (HUANG et al., 2017).

nais são mantidos conforme os da original, eles não são alterados e portanto o inicializador Xavier não é necessário neste caso.

4.3 Otimização utilizando computação evolutiva

Encontrar bons parâmetros para a arquitetura de uma CNN é um problema complexo que envolve testes empíricos, demanda tempo e experiência. Diversos trabalhos na literatura têm feito uso de técnicas de otimização para auxiliar na definição da arquitetura das CNNs. Entre as técnicas de otimização que podem ser utilizadas estão os algoritmos evolutivos AG e PSO.

Embora existam trabalhos que utilizam AGs ou PSOs para otimização de CNNs, nenhum deles aplica as técnicas para CNNs no domínio específico de imagens termográficas da mama. Com isto, propomos dois algoritmos, um AG e um PSO para escolha de boas arquiteturas e hiperparâmetros para a parte densamente conectada das CNNs escolhidas no problema de detecção do câncer de mama através de imagens termográficas. Nesta sessão, vamos detalhar os parâmetros a serem otimizados e os algoritmos do AG e do PSO propostos.

Para utilizar técnicas de otimização, faz-se necessário definir quais parâmetros serão otimizados. Para esta dissertação, foram escolhidos os seguintes parâmetros para otimização: taxa de aprendizado, número de camadas densamente conectadas, número de neurônios em cada camada, presença ou não de *dropout* e o valor da taxa quando *dropout* estiver presente. Além da definição destes parâmetros, faz-se necessário definir o domínio de cada parâmetro. Inspirados em diversos trabalhos da literatura e nas redes pré-treinadas que utilizamos, definimos os seguintes domínios:

- Taxa de aprendizado (LR): $LR = 10^{-lr}$, $lr \in [1, 6]$ e $lr \in \mathbb{Z}$
- Número de neurônios em cada camada (N_n): 2^{N_n} , $N_n \in [3, 10]$ e $N_n \in \mathbb{Z}$
- Taxa de dropout (t_d): $t_d \in [0, 0, 6]$ e $N_n \in \mathbb{R}$

Os parâmetros “número de camadas densamente conectadas” e “depois de qual camada há *dropout*” são parâmetros implícitos e não definidos como parâmetros da otimização visto que são consequência do indivíduo ou da partícula.

Tanto para o AG quanto para o PSO é necessário impor restrições para garantir CNNs válidas para o problema em questão. Algumas destas restrições são: a última camada densamente conectada necessariamente tem 2 neurônios de saída, já que temos um problema binário; não podem haver dois *dropouts* em sequência sem uma camada densamente conectada entre eles.

4.3.1 AG

O algoritmo do AG proposto está descrito no Algoritmo 1 e os detalhes são apresentados na sequência.

O primeiro passo para utilização do AG é a definição do indivíduo. Inspirados pelo indivíduo apresentado em (FIDELIS; LOPES; FREITAS, 2000) optamos por utilizar indivíduos de tamanho fixo que proporcionam simplicidade em operadores como o *crossover*, já que *crossovers* padrões da literatura podem ser utilizados, porém temos flexibilidade para adicionar ou remover uma camada densamente conectada (*DC*) ou um *dropout* D . Essa adição ou remoção de uma camada é realizada ligando ou desligando um bit. A Figura 14 mostra a estrutura de um indivíduo do AG. Os valores de LR e DC do indivíduo

Algoritmo 1 GA algorithm

```

1: population ← initializePopulation(populationSize)
2: fitness ← []
3: for i < populationSize do
4:   fitness[i] ← calcIndividualFitness(population[i])
5: end for
6: population, fitness ← sortDecreasing(population, fitness)
7: bestIndividual ← population[0]
8: bestFitness ← fitness[0]
9: iteration ← 0
10: for iteration < generations do
11:   selectedParents1 ← selectionTour(population, fitness, tourSize)
12:   selectedParents2 ← selectionTour(population, fitness, tourSize)
13:   childrenPopulation ← []
14:   for i < selectedParents1.length do
15:     child1, child2 ← crossover(selectedParents1[i], selectedParents2[i])
16:     childrenPopulation+ = child1
17:     childrenPopulation+ = child2
18:   end for
19:   childrenPopulation ← applyMutation(childrenPopulation)
20:   childrenFitness ← []
21:   for i < childrenPopulation.length do
22:     childrenFitness[i] ← calcIndividualFitness(childrenPopulation[i])
23:   end for
24:   childrenPopulation+ = bestIndividual
25:   childrenFitness+ = bestFitness
26:   childrenPopulation, childrenFitness ←
     sortDecreasing(childrenPopulation, childrenFitness)
27:   population, fitness ← orderedReinsertion(childrenPopulation, childrenFitness)
28:   bestIndividual ← population[0]
29:   bestFitness ← fitness[0]
30: end for

```

são apenas os expoentes das potências correspondentes. O valor da taxa de aprendizado da CNN correspondente ao indivíduo do AG é 10^{-LR} e o número de neurônios das camadas convolucionais é 2^{DC} .

Representação indivíduo do AG									
Tipo de camada	LR	DC ₁	D ₁	DC ₂	D ₂	...	DC ₅	D ₅	DC ₆
Está presente?	1	0 or 1	0 or 1	0 or 1	0 or 1	...	0 or 1	0 or 1	1
Valor	LR ∈ [1; 6]	FC ∈ [3; 10]	D ∈ [0; 0,6]	FC ∈ [3; 10]	D ∈ [0; 0,6]	...	FC ∈ [3; 10]	D ∈ [0; 0,6]	1

Figura 14 – Exemplo da estrutura do indivíduo do AG

Cada indivíduo do AG é composto por 12 cromossomos e cada cromossomo tem duas partes: o bit ‘está presente?’ e o valor. O primeiro cromossomo contém as informações da taxa de aprendizado e sempre terá o ‘está presente?’ como 1, assim como o cromossomo 12

(camada DC_6) que indica a última camada densamente conectada que sempre estará ativa e cujo valor sempre será 1 (gerando uma saída de 2 neurônios). Os demais cromossomos indicam as camadas densamente conectadas nas posições pares e os *dropouts* nas posições ímpares. Nota-se que não foi definido no indivíduo um parâmetro para número de camadas densamente conectadas, mas ele está implícito já que uma camada pode estar ‘ativa’ ou ‘inativa’ e portanto a quantidade de camadas ativas é o número de camadas densamente conectadas da CNN.

É importante mencionar que a camada de *dropout* sempre se refere a camada densamente conectada imediatamente anterior. Portanto, caso a camada DC da posição i esteja inativa, o *dropout* da posição $i + 1$ será desconsiderado. Com isso, impedimos que dois *dropouts* em sequência sejam gerados. Outra observação relevante é que como o DC_6 sempre está presente com valores fixos (1 tanto para o ‘está presente?’ quanto para o valor), esse cromossomo será adicionado apenas na conversão do indivíduo para a CNN correspondente e não estará explícito nos demais indivíduos apresentados. Isso foi realizado exclusivamente para simplificação do algoritmo.

Como o objetivo do AG envolve otimizar uma arquitetura de CNN e cada indivíduo representa uma CNN, o cálculo da *fitness* envolve treinar uma CNN para que ela possa ser avaliada. A *fitness* do nosso AG é o F1-score do conjunto de validação. Ao analisar as primeiras CNNs geradas percebeu-se que em alguns casos a melhor rede sugerida pelo AG tinha um número não decrescente de número de neurônios em cada camada. Como esse comportamento não é comumente observado na camada densamente conectada de CNNs optamos por utilizar um fator de penalização para redes com essa característica. Consequentemente, um indivíduo que corresponde a seguinte quantidade de neurônios em cada camada 256 -> 256 -> 1024 será penalizado, enquanto um indivíduo que corresponde a 1024 -> 1024 -> 256 não será penalizado. A *fitness* é descrita na Equação 6. O valor de penalização de 0,3 foi escolhido empiricamente (então consideramos apenas 0,7 da *fitness*).

$$fitness = \begin{cases} 0,7 * \text{F1-score da validação,} & \begin{array}{l} \text{número de neurônios} \\ \text{em cada camada é} \\ \textbf{não decrescente} \end{array} \\ \text{F1-score da validação,} & \begin{array}{l} \text{número de neurônios} \\ \text{em cada camada é} \\ \textbf{não crescente} \end{array} \end{cases} \quad (6)$$

Além da definição do indivíduo e da *fitness*, para o uso do AG é necessário definir quais métodos de seleção, *crossover*, mutação e reinserção serão utilizados. Na literatura diversos trabalhos que abordam otimização de CNNs por AG utilizam *crossover* de ponto único, portanto, esse foi o *crossover* escolhido. No *crossover* de um ponto sorteia-se uma posição randômica do indivíduo que vai ser o ponto de corte para a troca de material

genético. Desta forma, o primeiro filho recebe os genes das posições de 1 até ponto de corte de um dos pais (pai 1) e do ponto de corte em diante do outro (pai 2) e para o segundo filho o contrário, genes do pai 2 de 1 até o ponto de corte seguidos de gene do pai 1 do ponto de corte em diante. A Figura 15 ilustra como é realizado o *crossover*:



Figura 15 – Exemplo do crossover de ponto único

Para a **seleção** optamos por utilizar o torneio pela simplicidade em alterar a pressão seletiva do AG apenas alterando o valor do *tour*. O torneio consiste em formar subgrupos e entre os indivíduos deste subgrupo o que tem a melhor *fitness* é escolhido. O *tour* é o número de indivíduos que devem participar de cada subgrupo. Para a seleção de cada pai, torneio com *tour* indivíduos é realizado.

A **mutação** pode ser realizada de duas formas: no bit que ‘está presente?’ e nesse caso ocorre uma inversão de bit; ou no campo do valor efetivamente. Para mutação no valor, sorteia-se um novo valor no domínio do parâmetro. Pelos menos dois valores são escolhidos aleatoriamente para realizar cada mutação, um indicando qual cromossomo deve ser alterado e um indicando qual parte do cromossomo será alterado, caso a parte alterada seja o valor, gera-se então outro valor aleatório. A Figura 16 ilustra um exemplo de como é realizada a mutação. Neste exemplo, os valores em vermelho indicam os valores alterados pela mutação. Reforçamos que, conforme já mencionamos, o valor ‘está presente?’ da LR não pode ser alterado e portanto caso o cromossomo a ser alterado pela mutação for o primeiro, necessariamente a parte a ser alterada dele é o valor. A quantidade de cromossomos a ser alterada é um dos parâmetros do algoritmo, *tm*, que varia em diferentes experimentos.

Tipo de camada	LR	DC ₁	D ₁	DC ₂	D ₂	DC ₃	D ₃	DC ₄	D ₄	DC ₅	D ₅
Está presente?	1	0	1	1	1	1	1	0	1	1	1
Valor	3	10	0,30	7	0,13	9	0,31	8	0,23	12	0,11

↓

Tipo de camada	LR	DC ₁	D ₁	DC ₂	D ₂	DC ₃	D ₃	DC ₄	D ₄	DC ₅	D ₅
Está presente?	1	0	1	0	1	1	0	0	1	1	1
Valor	2	10	0,30	7	0,4	9	0,31	8	0,23	3	0,11

Figura 16 – Exemplo da mutação

Finalmente, optamos por utilizar a reinserção ordenada e elitismo mantendo o melhor indivíduo para a **próxima geração**. Desta forma o melhor indivíduo pai é mantido para a próxima geração, assim como os $tp-1$ melhores filhos onde tp é o tamanho da população.

4.3.2 PSO

O algoritmo do PSO proposto está descrito no Algoritmo 2 e os detalhes são apresentados na sequência.

Embora os parâmetros a serem otimizados pelo PSO sejam os mesmos do AG, diferente do indivíduo do AG, para a partícula do PSO optamos por uma partícula com tamanho variável inspirado no modelo proposto por (JUNIOR; YEN, 2019). Uma partícula do

Algoritmo 2 PSO

```

1:  $swarm \leftarrow initSwarm(swarmSize)$ 
2: for particle in swarm do
3:    $particle.fitness \leftarrow calcPositionFitness(particle.position)$ 
4: end for
5:  $gBest \leftarrow findGBest(swarm)$ 
6:  $iteration \leftarrow 0$ 
7: for  $i < iteration$  do
8:   for particle in swarm do
9:      $diffPbest \leftarrow calcParticleDiff(particle.position, particle.pBest.position)$ 
10:     $diffGbest \leftarrow calcParticleDiff(particle.position, gBest.position)$ 
11:     $newVelocity \leftarrow calcParticleVelocity(diffPbest, diffGbest, particle.pBest.position, gBest.position)$ 
12:     $particle.position \leftarrow updateParticlePosition(particle.position, newVelocity)$ 
13:     $particle.fitness \leftarrow calcPositionFitness(particle.position)$ 
14:    if  $particle.pBest.fitness < particle.fitness$  then
15:       $particle.pBest.fitness \leftarrow particle.fitness$ 
16:       $particle.pBest.position \leftarrow particle.position$ 
17:    end if
18:    if  $gBest.fitness < particle.fitness$  then
19:       $gBest.fitness \leftarrow particle.fitness$ 
20:       $gBest.position \leftarrow particle.position$ 
21:    end if
22:   end for
23: end for
24: return  $gBest$ 

```

PSO é composta por múltiplos fragmentos. Cada fragmento tem duas partes: tipo e valor. O tipo indica qual parte da CNN aquele fragmento representa. Os tipos que consideramos podem ser LR (taxa de aprendizado), DC (densamente conectada) e D (*dropout*). O campo do valor indica o valor daquele fragmento, respeitando o domínio de cada parâmetro conforme definido anteriormente. De forma similar ao que acontece no AG, na conversão de uma partícula para a CNN correspondente o valor da LR é utilizada como 10^{-LR} e o valor DC como 2^{DC} . O último fragmento da partícula é um fragmento de DC com o valor 1 para garantir que a última camada seja uma camada densamente conectada com 2 neurônios, equivalente ao que acontece no AG, esse último fragmento é apenas adicionado quando há a conversão da partícula para a CNN para fins de simplicidade do algoritmo. Há ainda a restrição de que dois fragmentos de D não podem ser sequenciais. A Figura 17 ilustra um exemplo de partícula.

A Figura 18 ilustra uma comparação entre um indivíduo do AG e uma partícula do PSO que representam a mesma CNN.

A avaliação de cada partícula é realizada conforme a Equação 6, a mesma *fitness* apresentada para o AG. O valor da *fitness* da partícula é o F1-score do conjunto de validação aplicando ou não a penalização a depender do comportamento não crescente ou

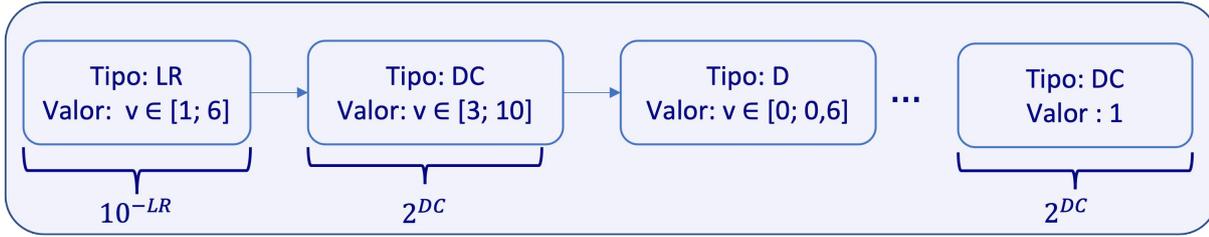


Figura 17 – Definição da partícula do PSO.

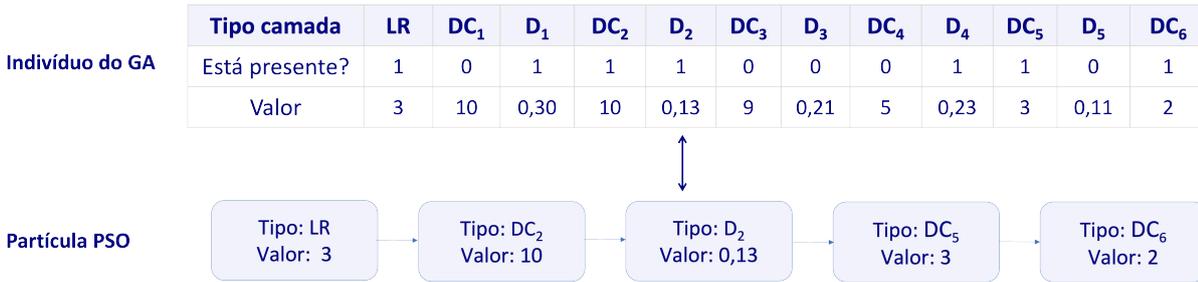


Figura 18 – Comparação entre uma partícula do PSO e um indivíduo do AG que representam uma mesma CNN.

não decrescente do número de neurônios em cada camada (Equação 6).

Além disto, o PSO envolve cálculo da velocidade e cálculo da diferença entre partículas que também precisam ser definidos. O cálculo da diferença entre duas partículas é realizado fragmento por fragmento inspirado pelo trabalho de (JUNIOR; YEN, 2019). A diferença entre os fragmentos é definida como:

- 0: quando o tipo dos fragmentos são iguais;
- o fragmento da primeira partícula: caso os fragmentos sejam diferentes ou quando a primeira partícula tem fragmentos a mais que não tem correspondentes na segunda;
- -1: quando a segunda partícula tem um fragmento a mais que não tem correspondente na primeira partícula.

A Figura 19 ilustra dois exemplos de como a diferença entre duas partículas é calculada. A diferença gera um conjunto de fragmentos, embora não seja um nova partícula válida.

O cálculo da velocidade da partícula, também inspirado pelo trabalho de (JUNIOR; YEN, 2019), é realizado fragmento por fragmento. A velocidade de cada fragmento é determinada pela diferença entre $gBest - P$ ou $pBest - P$ onde $gBest$ é o melhor global (entre todas as partículas), $pBest$ o melhor da partícula e P é a partícula. A escolha entre a primeira diferença ($gBest - P$) ou a segunda ($pBest - P$) é realizada a partir de r , um

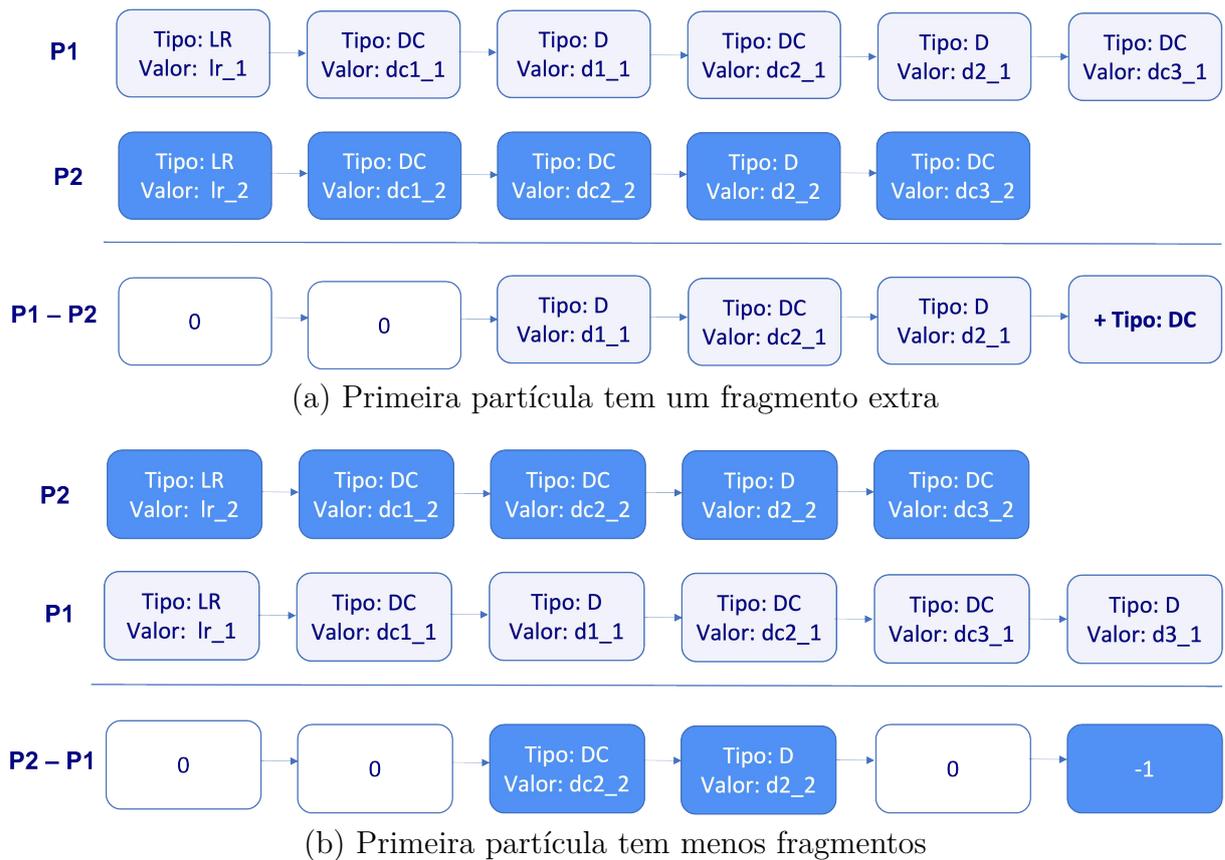


Figura 19 – Cálculo da diferença de duas partículas

valor aleatório entre $[0, 1]$, e de um parâmetro Cg (também entre $[0, 1]$) definido no início da execução do algoritmo com a Equação 7:

$$\text{velocidade do fragmento} = \begin{cases} p\text{Best} - P & \text{se } r > Cg \\ g\text{Best} - P & \text{caso contrário} \end{cases} \quad (7)$$

A Figura 20 ilustra um exemplo do cálculo da velocidade onde o $Cg=0,5$ e o r são os valores indicados entre os dois fragmentos.

Finalmente é preciso atualizar a posição da partícula dada sua velocidade. A atualização da posição pode manter, remover ou alterar os fragmentos da partícula a depender da velocidade conforme:

- ❑ Se a velocidade do fragmento = 0, então o fragmento da partícula é mantido;
- ❑ Se a velocidade do fragmento = fragmento do tipo DC ou D, então o fragmento é substituído pelo fragmento da velocidade;
- ❑ Se a velocidade do fragmento = -1, então o fragmento da partícula é removido.

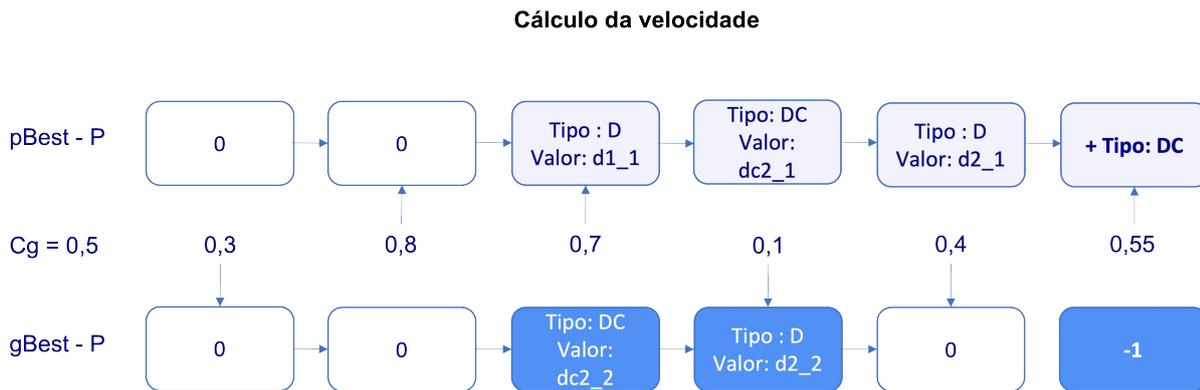


Figura 20 – Exemplo cálculo da velocidade de uma partícula.

A Figura 21 contém um exemplo de atualização da posição. Considere $P1_t$ a posição inicial da partícula e $P1_{t+1}$ a posição final da partícula depois da atualização.



Figura 21 – Exemplo atualização da posição de uma partícula.

4.4 *Surrogate*

Devido ao custo computacional e ao tempo gasto para calcular a *fitness* tanto do AG quanto do PSO, que envolve treinar CNNs, decidimos utilizar um modelo *surrogate*. O modelo *surrogate* escolhido foi um modelo baseado em *random forest*. A *random forest* foi escolhida por apresentar bons resultados mesmo com quantidades limitadas de dados para treinamento (SUN et al., 2019a).

O modelo da *random forest* é treinado com uma arquitetura de CNN (que está codificada no indivíduo do AG ou na partícula do PSO) e o respectivo F1-score correspondente dessa arquitetura. O modelo proposto utiliza a abordagem *online* e a combinação de *fitness* real com *fitness* do *surrogate* baseada em indivíduos. Nos algoritmos evolutivos, as duas primeiras gerações/iterações são inteiramente calculadas com a *fitness* real. No nosso problema, isso significa treinar a CNN. Isso ocorre para podermos gerar dados para treinar o modelo da *random forest*. Nas gerações posteriores, metade dos indivíduos são

treinados com a *fitness* real e a outra metade com a *fitness* do *surrogate*. A metade treinada com a *fitness* real é então utilizada para retreinar o modelo com todos os dados disponíveis até o momento (todos os dados cujos F1-score foram calculados com a *fitness* real). Desta forma, quando a outra metade dos indivíduos tiverem suas *fitness* calculadas, o modelo já estará atualizado. O Algoritmo 3 descreve o algoritmo utilizado.

Algoritmo 3 Cálculo fitness com surrogate

```

1: fitness ← []
2: trainSurrogateData ← []
3: if iteração < 2 then
4:   for individuo in individuos do
5:     fitnessIndividuo ← calcRealFitness(individuo)
6:     fitness+ = fitnessIndividuo
7:   end for
8:   if iteração == 1 then
9:     encodedIndividuos ← encodeIndividuos(individuos)
10:    surrogateModel ← createSurrogateModel(encodedIndividuos, fitness)
11:   end if
12:   trainSurrogateData ← individuos
13: else
14:   i ← 0
15:   for i < (individuos.length/2) do
16:     fitnessIndividuo ← calcRealFitness(individuo[i])
17:     fitness+ = fitnessIndividuo
18:   end for
19:   trainSurrogateData ← individuos[0]...individuos[individuos.length/2]
20:   encodedIndividuos ← encodeIndividuos(trainSurrogateData)
21:   surrogateModel ← createSurrogateModel(encodedIndividuos, fitness)
22:   i ← (individuos.length/2)
23:   for i < individuos.length do
24:     fitnessIndividuo ← calcFitnessSurrogate(surrogateModel, individuo[i])
25:     fitness+ = fitnessIndividuo
26:   end for
27: end if
28: return fitness, trainSurrogate

```

A primeira etapa para utilizar a *random forest* é definir como a conversão do indivíduo do AG e da partícula do PSO para a entrada da *random forest* será realizada. Chamamos essa etapa de *encode*. Para a utilização da *random forest*, todas as entradas devem ser do mesmo tamanho, no AG isso já acontece, porém no PSO a partícula tem tamanho variável. Desta forma, a primeira etapa do *encode*, que só acontece para o PSO, é converter a partícula do PSO para uma estrutura de tamanho fixo equivalente a do AG, com a *flag* estaPresente indicando se aquela camada está ou não ativa e o valor. A Figura 22 mostra um exemplo de como a conversão é feita.

A próxima etapa do *encode* é converter o indivíduo do AG na entrada da *random*

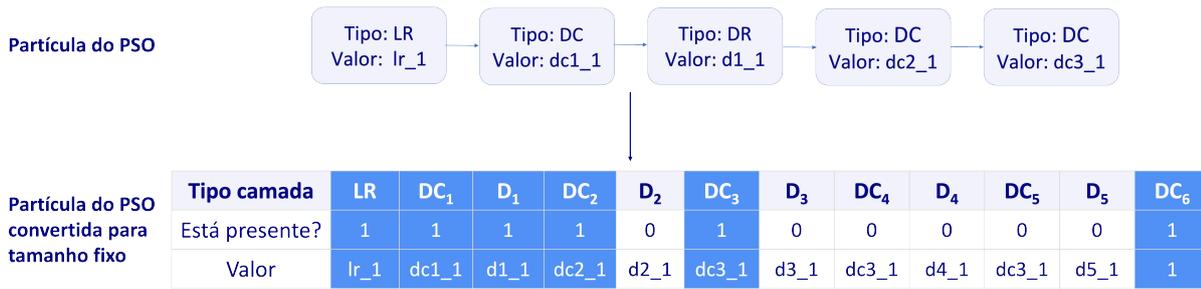


Figura 22 – Exemplo conversão de uma partícula do PSO para tamanho fixo.

forest. A entrada da *random forest* é um *array* de 1 dimensão com 36 posições (12*3). 12 equivale ao número de genes de um indivíduo (1 * LR + 6 * DC + 5 * D). Além disso, cada gene do indivíduo é convertido para três partes (tipo, estaPresente, valor). O tipo é um valor entre 0 e 2, onde 0 indica LR, 1 indica DC e 2 indica D, o estaPresente e o valor tem a mesma interpretação do indivíduo. A Figura 23 mostra um exemplo de um indivíduo e o respectivo *encode* dele que é a entrada da *random forest*.

Tipo camada	LR	DC ₁	D ₁	DC ₂	D ₂	DC ₃	D ₃	DC ₄	D ₄	DC ₅	D ₅
Está presente?	1	0	1	1	1	1	1	0	1	1	1
Valor	3	10	0,30	10	0,13	8	0,31	5	0,23	3	0,11

Tipo camada, está presente, valor	0; 1; 3	1; 0; 10	2; 1; 0,30	1; 1; 10	2; 1; 0,13	1; 1; 8	2; 1; 0,31	1; 0; 5	2; 1; 0,23	1; 1; 3	2; 1; 0,11	1; 1; 1
-----------------------------------	---------	----------	------------	----------	------------	---------	------------	---------	------------	---------	------------	---------

Figura 23 – Exemplo conversão indivíduo para a entrada da *random forest*.

Após o *encode*, a *random forest* é então treinada com esses dados e os correspondentes F1-scores. O modelo obtido é então utilizado para o cálculo das *fitness* quando apropriado. Diferente da *fitness* real apresentada na Equação 6, a *fitness* calculada com o modelo *surrogate* não tem a penalização para casos onde o número de neurônios é não decrescente. A penalização não foi necessária porque não houve entre os experimentos realizados, a seleção de melhor rede com arquitetura onde o número de neurônios é não decrescente.

Experimentos e Discussões

Neste capítulo, serão detalhados os experimentos realizados e os resultados obtidos com as respectivas implicações destes resultados. Para o desenvolvimento deste trabalho utilizamos Python, PyTorch e MATLAB. Além disto, fizemos uso do Google Colab para utilizar as GPUs (*graphics processing units*) disponibilizadas pela ferramenta.

Em todos os experimentos apresentados foi utilizado o otimizador Adam. Além disto, todos os experimentos tiveram os dados divididos em três conjuntos: treinamento, teste e validação com proporções de 70%, 15% e 15% respectivamente. Os dados foram divididos de forma que, em cenários onde há mais de uma imagem por paciente, todas as imagens deste mesmo paciente ficarão em apenas um dos conjuntos (treinamento, validação ou teste). Por exemplo, se o paciente 1 tem 5 imagens, todas as 5 imagens estarão apenas no grupo de treinamento.

5.1 Métricas

Para avaliar as CNNs, utilizamos as seguintes métricas: acurácia, sensibilidade, especificidade, F1-score.

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \quad (8)$$

$$SEN = \frac{TP}{TP + FN} \quad (9)$$

$$SPE = \frac{TN}{TN + FP} \quad (10)$$

$$Prec = \frac{TP}{TP + FP} \quad (11)$$

$$F1 - Score = \frac{2 * Prec * SEN}{Prec + SEN} \quad (12)$$

onde TP é verdadeiro positivo, FP é falso positivo, TN é verdadeiro negativo e FN é falso negativo.

5.2 Experimentos manuais

Chamamos experimentos manuais os experimentos em que foram testados 6 diferentes arquiteturas para a parte densamente conectada (DC) da CNN escolhidas empiricamente e baseado nas arquiteturas originais das CNNs. Os testes foram realizados com arquiteturas de uma, duas e três camadas conforme a Tabela 2:

Tabela 2 – Tipo do experimento e a arquitetura correspondente para a camada densamente conectada da CNN.

Tipo do Experimento	Arquitetura da parte densamente conectada (DC) da rede
(1)	1 DC
(2)	2 DC com 256 neurônios como saída para a primeira camada e entrada na segunda (256 como saída da primeira camada e 2 como saída da segunda camada)
(3)	2 DC com 512 neurônios (512, 2)
(4)	2 DC com 1024 neurônios (1024, 2)
(5)	3 DC com 4096 e 1024 neurônios (4096, 1024, 2)
(6)	3 DC com 4096 neurônios (4096, 4096, 2)

Além disso, nesses experimentos, duas taxas de aprendizado (LR) foram testadas: 0,0001 e 0,001. Para o número de épocas, consideramos 30 épocas com e sem *early stopping*. O uso do *early stopping*, quando ativo, faz com que após 10 épocas sem redução no erro do conjunto de validação o treinamento seja finalizado. Desta forma, evitamos que as redes sofram o chamado *overfitting*, i.e., quando uma rede aprende as especificidades dos dados nos quais é treinada e acaba não sendo generalizável.

Utilizamos a transferência de aprendizado, desta forma, a parte convolucional da rede é mantida conforme a CNN original e seus pesos não são atualizados.

Foram realizados testes com as redes descritas acima com quatro subconjuntos de dados:

- grupo-1: com o mesmo número de pacientes reportado por (CABIOĞLU; OĞUL, 2020) (147 pacientes saudáveis e 34 pacientes doentes). Destes pacientes foram consideradas imagens de todas as poses disponíveis (em geral 5 imagens por paciente), totalizando 722 imagens de pacientes saudáveis e 169 imagens de pacientes doentes. Além disto, para lidar com o desbalanceamento da base foi utilizado aumento

de dados em ambos os conjuntos, porém com proporções diferentes, resultando em 2796 imagens, sendo 1444 saudáveis e 1352 doentes. Os grupos continuaram ligeiramente desbalanceados porque optamos por aumentar na mesma proporção todos os pacientes de cada classe, não gerando mais imagens para alguns dos pacientes do que os outros na mesma classe. Neste grupo o pré-processamento utilizado foi o JetColormap.

- grupo-2: os mesmos pacientes do grupo-1 porém com apenas as imagens frontais (logo 1 imagem por paciente). Neste caso, também lançou-se mão do aumento de dados em ambos os conjuntos com as mesmas proporções utilizadas para o grupo-1. Logo, obteve-se 294 imagens saudáveis e 272 imagens de pacientes doentes. Neste grupo o pré-processamento utilizado também foi o JetColormap.
- grupo-3: os mesmos pacientes do grupo-2 porém com aumento de dados apenas nos doentes. Logo, obteve-se 147 imagens saudáveis e 135 imagens de pacientes doentes. Neste grupo o pré-processamento utilizado novamente foi o JetColormap.
- grupo-4: utilizando 38 pacientes de cada classe (base balanceada) e sem aumento de dados, com o pré-processamento do min-max.

Os grupos 1, 2 e 3 foram inspirados no número de pacientes e imagens e forma de pré-processar as imagens de (CABIOĞLU; OĞUL, 2020). Já grupo-4 foi escolhido como um grupo que utiliza todas as imagens frontais de pacientes doentes, 38 imagens, e mantém a base balanceada. Portanto, também com 38 imagens saudáveis. A forma de pré processamento foi selecionada por ser a mais simples possível.

Cada um dos grupos foi testado com todas as 12 diferentes configurações, 6 diferentes arquiteturas e cada uma destas testada com cada uma das duas taxas de aprendizado. Além disso, em todas elas foi utilizado o *early stopping*. As Tabelas 3 - 8 apresentam os melhores resultados obtidos no conjunto de teste das CNNs com os quatro subconjuntos de dados utilizados. Apresentamos apenas resultados cujas métricas consideradas apresentaram valores superiores a 70%. Nessa etapa de experimentos manuais, apenas os resultados do grupo-3 utilizaram o inicializador Xavier, para os demais grupos essa forma de inicializar os pesos não foi utilizada.

Na Tabela 3 são apresentados os resultados obtidos com o grupo-1. Com esses dados, o melhor resultado obtido foi 0,7655 de F1-score e apenas a DenseNet apresentou resultados significativos (acima de 0,7), conforme pode ser observado na Tabela. Como o grupo-1 tem todas as imagens de um dado paciente e existem imagens laterais onde apenas uma das mamas é visível, a nossa hipótese é que isto pode ser um problema para classificação porque é possível que seja apresentado a rede uma imagem de uma mama saudável porém de um paciente doente, i.e., o câncer está na mama que não foi apresentada. Isso ocorre devido a classificação entre doente e saudável ser realizada por paciente e não por mama.

Então o câncer pode estar em qualquer uma das mamas e a informação de em qual mama está o câncer não está disponível. Por exemplo, quando o câncer estiver na mama direita e a rede for treinada com a imagem lateral esquerda na qual apenas a mama esquerda está visível, a rede será treinada com uma imagem saudável para a classe de doente. Além disso, dependendo de onde o tumor está localizado, ele pode não estar visível na imagem lateral. Para validar essa hipótese, testamos com o grupo-2, onde os mesmos pacientes são utilizados porém apenas com as imagens frontais.

Tabela 3 – Resultados ‘manuais’ obtidos com o grupo-1 - todas as poses.

LR	Tipo do Experimento	CNN utilizada	ACC (%)	SE (%)	SPE (%)	F1-Score
0,0001	(3)	DenseNet	74,57	75,42	73,66	0,7542
	(5)	DenseNet	75,00	71,67	78,57	0,7478
0,001	(1)	DenseNet	73,71	73,33	74,11	0,7426
	(5)	DenseNet	77,16	72,08	82,59	0,7655

Os resultados obtidos com o grupo-2 são apresentados nas Tabelas 4 e 5. A Tabela 4 tem os resultados obtidos com a taxa de aprendizado 0,001 e a Tabela 5 tem os resultados obtidos com a LR 0,0001. O melhor resultado com os dados do grupo-2 foi também com a DenseNet com 0,8824 de F1-score com a LR 0,0001. Diferente dos resultados apresentados anteriormente, considerando esse subconjunto dos dados, obtivemos resultados significativos também com a VGG e a ResNet, sendo que o melhor resultado com a VGG foi de 0,7692 de F1-score e para a ResNet F1-score de 0,8381. Comparando esses resultados com os anteriores percebemos que houve melhora significativa quando utilizando apenas as imagens frontais, corroborando a nossa hipótese apresentada anteriormente. Temos então que as imagens laterais podem impactar negativamente a classificação quando as classes são dadas por paciente e não por mama. No grupo-2, conforme já mencionamos, optamos por aplicar o aumento de dados em ambas as classes buscando um banco de dados não enviesado. Comparando os resultados obtidos com os dados do grupo-2 e os resultados apresentado em (CABIOĞLU; OĞUL, 2020), já que ambos utilizaram o mesmo número de pacientes, percebemos que os nossos resultados são inferiores. Acreditamos que é possível que o aumento apenas do conjunto dos dados doentes gerem resultados ainda melhores, embora entendemos que está não é a melhor abordagem.

Buscando uma comparação mais equivalente com o que foi feito no trabalho de (CABIOĞLU; OĞUL, 2020), optamos por rodar as CNNs com os mesmos números de pacientes utilizados por eles com aumento apenas dos pacientes doentes. Esse conjunto de dados

Tabela 4 – Resultados ‘manuais’ obtidos com o grupo-2 - LR 0,001.

LR	Tipo do Experimento	CNN utilizada	ACC (%)	SE (%)	SPE (%)	F1-Score
0,001	(1)	DenseNet	80,85	79,17	82,61	0,8085
	(2)	VGG	75,53	70,83	80,43	0,7473
	(3)	VGG	77,66	72,92	82,61	0,7692
	(4)	VGG	74,47	70,83	78,26	0,7391
		DenseNet	80,85	79,17	82,61	0,8085
	(5)	DenseNet	84,04	87,50	80,43	0,8485
		ResNet	81,91	87,50	76,09	0,8317

Tabela 5 – Resultados ‘manuais’ obtidos com o grupo-2 - LR 0,0001.

LR	Tipo do Experimento	CNN utilizada	ACC (%)	SE (%)	SPE (%)	F1-Score
0,0001	(1)	VGG	76,60	72,92	80,43	0,7609
		DenseNet	80,85	79,17	82,61	0,8085
		ResNet	80,85	83,33	78,26	0,8163
	(2)	DenseNet	77,66	70,83	84,78	0,7640
		ResNet	77,66	75,00	80,43	0,7742
	(3)	ResNet	79,79	79,17	80,43	0,8000
	(4)	DenseNet	87,23	93,75	80,43	0,8824
		ResNet	81,91	91,67	71,74	0,8381
	(5)	VGG	75,53	75,00	76,09	0,7579

é chamado de grupo-3. As Tabelas 6 e 7 apresentam os resultados obtidos com esses dados. Conforme era esperado os resultados melhoram significativamente sendo que os melhores resultados foram obtidos com a DenseNet com F1-score de 0,9167 com ambas as LR. Além disto, a VGG também apresentou resultados melhores com F1-score de 0,8696 e a ResNet com 0,9130. Portanto, para todas as CNNs testadas o grupo-3 apresentou melhores resultados que os anteriores, porém ainda ligeiramente inferiores ao 0,94 reportado por (CABIOĞLU; OĞUL, 2020). Embora tenhamos utilizado o mesmo número de pacientes, não sabemos quais pacientes especificamente foram utilizados pelos autores e também não utilizamos as mesmas CNNs, já que os autores utilizam a AlexNet, o que justifica está ligeira diferença.

Tabela 6 – Resultados ‘manuais’ obtidos com o grupo-3 - LR 0,001.

LR	Tipo do Experimento	CNN utilizada	ACC (%)	SE (%)	SPE (%)	F1-Score
0,001	(1)	VGG	82,98	83,33	82,61	0,8333
		DenseNet	85,11	79,17	91,30	0,8444
		ResNet	91,49	87,50	95,65	0,9130
	(2)	VGG	87,23	83,33	91,30	0,8696
		DenseNet	87,23	79,17	95,65	0,8636
		ResNet	82,98	87,50	78,26	0,8400
	(3)	VGG	85,11	83,33	86,96	0,8511
		ResNet	82,98	91,67	73,91	0,8462
	(4)	VGG	82,98	79,17	86,96	0,8261
		DenseNet	91,49	91,67	91,30	0,9167
		ResNet	87,23	87,50	86,96	0,8750
	(5)	VGG	85,11	91,67	78,26	0,8627
		DenseNet	87,23	91,67	82,61	0,8800
		ResNet	91,49	87,50	95,65	0,9130
	(6)	VGG	87,23	79,17	95,65	0,8636
		DenseNet	87,23	91,67	82,61	0,8800
		ResNet	87,23	87,50	86,96	0,8750

Tabela 7 – Resultados ‘manuais’ obtidos com o grupo-3 - LR 0,0001.

LR	Tipo do Experimento	CNN utilizada	ACC (%)	SE (%)	SPE (%)	F1-Score
0,0001	(1)	VGG	87,23	79,17	95,65	0,8636
		DenseNet	85,11	83,33	86,96	0,8511
		ResNet	85,11	87,50	82,61	0,8571
	(2)	VGG	85,11	79,17	91,30	0,8444
		DenseNet	87,23	91,67	82,61	0,8800
		ResNet	85,11	83,33	86,96	0,8511
	(3)	VGG	82,98	79,17	86,96	0,8261
		DenseNet	87,23	79,17	95,65	0,8636
		ResNet	80,85	95,83	65,22	0,8364
	(4)	VGG	87,23	83,33	91,30	0,8696
		DenseNet	89,36	83,33	95,65	0,8889
		ResNet	85,11	75,00	95,65	0,8372
	(5)	VGG	85,11	70,83	100,00	0,8293
		DenseNet	91,49	91,67	91,30	0,9167
		ResNet	87,23	87,50	86,96	0,8750
	(6)	VGG	85,11	79,17	91,30	0,8444
		DenseNet	85,11	79,17	91,30	0,8444
		ResNet	91,49	91,67	91,30	0,9167

Finalmente, optamos por testar com o conjunto de dados do grupo-4 onde a base é balanceada com 38 pacientes em cada classe e também apenas com as imagens frontais. A Tabela 8 apresenta os resultados significativos obtidos. Percebemos que o melhor resultado obtido entre todos os experimentos manuais foi obtido com o grupo-4 com a rede DenseNet que obteve 0,92 de F1-score. Ainda com o grupo-4 a ResNet obteve 0,83 de F1-score e a VGG não apresentou resultados significativos (os resultados obtidos foram abaixo de 70% para as métricas consideradas).

Os resultados obtidos com os experimentos manuais superam os resultados de alguns dos trabalhos apresentados anteriormente, entre eles, (FAROOQ; CORCORAN, 2020),

Tabela 8 – Resultados ‘manuais’ obtidos com o grupo-4.

LR	Tipo do Experimento	CNN utilizada	ACC (%)	SE (%)	SPE (%)	F1-Score
0,001	(1)	DenseNet	83,33	83,33	83,33	0,83333
	(3)	ResNet	83,33	83,33	83,33	0,83333
	(6)	DenseNet	91,67	100,00	83,33	0,92308

(KIYMET et al., 2019) e (CHAVES et al., 2020). Além disto, com esses experimentos obtivemos resultados equivalente a (ZULUAGA-GOMEZ et al., 2019) e ligeiramente inferiores a (TORRES-GALVÁN et al., 2021). Nossos resultados foram ainda inferiores aos apresentados em (CABIOĞLU; OĞUL, 2020), (BAFFA; LATTARI, 2018), (ROSLIDAR et al., 2019) e (MAMBOU et al., 2018).

Esses resultados serão utilizados como base para as comparações com os resultados obtidos pela otimização que serão apresentados na próxima seção. Nas Tabelas 9, 10, 11 e 12, é apresentado um resumo com os melhores resultados de cada rede para os quatro grupos avaliados (grupo-1 grupo-2, grupo-3 e grupo-4 respectivamente). Nessas tabelas com os resumos dos resultados, os resultado da ResNet e VGG para o grupo-1 e da VGG para o grupo-4 não haviam sido apresentados anteriormente devido a não atingirem os limites de valores mínimos obtidos para exibição nas tabelas. Entretanto, como essa informação será utilizada para comparação com as demais técnicas, esses dados foram explicitados nas tabelas resumo.

Tabela 9 – Resumo grupo-1.

LR	Tipo do Experimento	CNN utilizada	ACC (%)	SE (%)	SPE (%)	F1-Score
0,001	(5)	DenseNet	77,16	72,08	82,59	0,7655
0,0001	(4)	ResNet	73,92	79,16	68,30	0,7584
0,001	(5)	VGG	70,04	62,50	78,12	0,6833

Tabela 10 – Resumo grupo-2.

LR	Tipo do Experimento	CNN utilizada	ACC (%)	SE (%)	SPE (%)	F1-Score
0,0001	(4)	DenseNet	87,23	93,75	80,43	0,8824
0,0001	(4)	ResNet	81,91	91,67	71,74	0,8381
0,001	(3)	VGG	77,66	72,92	82,61	0,7692

Tabela 11 – Resumo grupo-3.

LR	Tipo do Experimento	CNN utilizada	ACC (%)	SE (%)	SPE (%)	F1-Score
0,001	(4)	DenseNet	91,49	91,67	91,30	0,9167
0,0001	(6)	ResNet	91,49	91,67	91,30	0,9167
0,001	(2)	VGG	87,23	83,33	91,30	0,8696

Tabela 12 – Resumo grupo-4.

LR	Tipo do Experimento	CNN utilizada	ACC (%)	SE (%)	SPE (%)	F1-Score
0,001	(6)	DenseNet	91,67	100,00	83,33	0,92308
0,001	(3)	ResNet	83,33	83,33	83,33	0,83333
0,001	(4)	VGG	75,0	50,0	100	0,6666

5.3 Experimentos com otimização

Nesta seção serão apresentados os experimentos realizados com as duas técnicas de otimização propostas: AG e PSO. Como em ambos os algoritmos a cada iteração é necessário treinar várias CNNs para o cálculo da *fitness* para múltiplos indivíduos/partículas, utilizamos paralelismo para esse cálculo. Desta forma, vários indivíduos/partículas podem ser avaliados ao mesmo tempo. Todo o restante dos algoritmos é sequencial, sendo o paralelismo utilizado exclusivamente para auxiliar no cálculo da *fitness*. Outro mecanismo utilizado para diminuir o custo computacional dos algoritmos é o *cache*. Cada indivíduo/partícula que é avaliado tem a arquitetura da CNN e o valor da *fitness* armazenados em um *cache*, desta forma, antes de treinar uma nova CNN é verificado se alguma CNN com aquela configuração já foi treinada previamente e, em caso afirmativo, o valor da *fitness* é recuperado do *cache* evitando o recálculo.

Conforme mencionado anteriormente, para a *fitness* utilizamos o F1-score do conjunto de validação. Entretanto, quando o AG ou PSO terminam as gerações/iterações, utilizamos o melhor indivíduo/partícula obtido e treinamos algumas CNN com aquela arquitetura para então avaliar o conjunto de teste. Para fazer esta avaliação do conjunto de teste, quatro CNNs com as configurações definidas pela otimização são treinadas variando apenas o número de épocas. São testadas CNNs com 30 e 50 épocas, ambas com e sem *early stopping* de 10 épocas. Todos os experimentos utilizam o inicializador Xavier.

Tanto nos experimentos com o AG quanto com o PSO, cada configuração diferente dos parâmetros é avaliada 5 vezes e os melhores resultados de cada uma dessas 5 rodadas serão avaliados. Sobre os grupos de dados testados na otimização, apenas o grupo-1 (grupo dos dados que considera imagens de todas as posições disponíveis) não foi avaliado na otimização com AG e PSO devido aos resultados apresentados serem muito inferiores aos demais grupos e também a grande quantidade de dados quando comparados aos outros grupos, essa quantidade de dados na otimização acabaria sendo muito custosa computacionalmente, podendo até mesmo se tornar inviável. Desta forma, os grupos grupo-2, grupo-3 e grupo-4 foram avaliados e testados com as técnicas de otimização.

5.3.1 AG

Para todos os experimentos realizados com o AG, utilizamos uma taxa de *crossover* (tr) de 80%. Essa taxa não significa que 80% dos pares de pais sofrerão *crossover*, mas sim que cada par de pais tem 80% de chance de sofrer o *crossover*. Desta forma, para cada par de pais selecionado sorteia-se um número aleatório entre $[0, 1]$, caso o número seja menor ou igual a 0,8 o *crossover* de um ponto é realizado, do contrário, os indivíduos são mantidos. Outro parâmetro que não foi alterado entre os diferentes experimentos foi o valor do *tour*. Utilizou-se apenas o *tour* de 2 para o torneio. Para o número de gerações também não houve alteração entre os diferentes experimentos, sendo que foram

testadas 10 gerações. Esses valores não foram variados porque experimentos prévios indicaram que não houve melhorias significativas que justificassem o custo de testar ainda mais combinações envolvendo esses quatro parâmetros (TR , $tour$, número de gerações e $crossover$ de um ou de dois pontos).

Entre os diferentes experimentos foram alteradas a taxa de mutação (TM), onde os seguintes valores foram testados: 20%, 30%, 40% e 60% e tamanho da população (TP) cujos seguintes valores foram avaliados: 10 e 20. Portanto, para cada subconjunto dos dados (grupos descritos anteriormente) 8 experimentos com o AG foram realizados, cada uma das 4 TM com cada um dos TP considerados. Além disso, lembramos que cada um dos 8 experimentos foram executados 5 vezes e cada uma das 5 vezes foi testada com 4 diferentes configurações para o número de épocas após o término da execução do AG.

As Tabelas 13, 14 e 15 mostram os melhores resultados obtidos com o AG para os grupos: grupo-2, grupo-3 e grupo-4 respectivamente (reforçamos que, conforme já mencionado, o grupo-1 não foi avaliado devido aos resultados muito inferiores aos demais e ao custo computacional de avaliar esse grupo). Nas tabelas, a coluna ‘Melhor Fitness AG’ se refere a *fitness* do F1-score do conjunto de validação do melhor indivíduo do AG, o NE é o número de épocas que gerou o melhor resultado no conjunto de teste, entre as 4 possibilidades mencionadas, ACC, SE, SP e F1-score se referem a acurácia, sensibilidade, especificidade e F1-score do conjunto de teste.

Tabela 13 – Melhores resultados com AG - grupo-2.

CNN	TP	TM	Melhor Fitness AG	NE	ACC (%)	SE (%)	SP (%)	F1-score
DenseNet	20	20	0,7407	30 com ES	87,23	93,75	80,43	0,8823
ResNet	10	40	0,6341	30	82,97	91,66	73,91	0,8461
VGG	20	40	0,7407	30	84,04	93,75	73,91	0,8571

Tabela 14 – Melhores resultados com AG - grupo-3.

CNN	TP	TM	Melhor Fitness GA	NE	ACC (%)	SE (%)	SP (%)	F1-score
DenseNet	10	40	0,8947	50 com ES	91,49	91,67	91,30	0,9167
ResNet	10	20	0,8235	50	91,49	87,50	95,65	0,9130
VGG	20	60	0,7619	30	89,36	95,83	82,61	0,9020

Tabela 15 – Melhores resultados com AG - grupo-4.

CNN	TP	TM	Melhor Fitness AG	NE	ACC (%)	SE (%)	SP (%)	F1-score
DenseNet	20	30	0,8571	30	91,66	100	83,33	0,9230
ResNet	20	60	0,9230	50	91,66	83,33	100	0,9090
VGG	10	30	0,9090	30	91,66	100	83,33	0,9230

A arquitetura e os hiperparâmetros das CNNs que geraram esses resultados apresentados na Tabela 13 que utilizam o grupo-2 são:

- ❑ DenseNet: LR 10^{-4} ; camada DC (4096, 256, *dropout* 0,3401, 2);
- ❑ ResNet: LR 10^{-5} ; camada DC (1024, 2);
- ❑ VGG: LR 10^{-3} ; camada DC (1024, 8, 2);

Para os dados apresentados na Tabela 14, que utilizaram o grupo-3, as arquitetura e os hiperparâmetros das CNNs são:

- ❑ DenseNet: LR 10^{-3} ; camada DC (256, *dropout* 0,2832, 8, *dropout* 0,2946, 2);
- ❑ ResNet: LR 10^{-4} ; camada DC (1028, *dropout* 0,5119, 2);
- ❑ VGG: LR 10^{-2} ; camada DC (2);

Já as arquiteturas correspondentes aos resultados da Tabela 15 que utilizam o grupo-4 são:

- ❑ DenseNet: LR 10^{-4} ; camada DC (2);
- ❑ ResNet: LR 10^{-5} ; camada DC (128, *dropout* 0,4267, 32, 2);
- ❑ VGG: LR 10^{-2} ;camada DC (32, *dropout* 0,3065, 8, *dropout* 0,3153, 2);

O AG com grupo-2 teve seus melhores resultados obtidos com o 30 épocas sem o *early stopping* (ES) na evolução do AG. Já para o grupo-3 e o grupo-4 os melhores resultados foram obtidos com ES, ou seja, 30 épocas com ES. Outra diferença entre os diferentes grupos de dados é o fator de penalização da *fitness*. Para o grupo-2, o fator de penalização utilizado foi de 0,6, já para os demais dois grupos a penalização foi de 0,7 conforme demonstrado na Equação 6. Esses fatores de penalização e quais os mais adequados para cada grupo de dados foram escolhidos empiricamente.

Analisando os resultados percebe-se que as maiores melhorias obtidas com a utilização do AG foram obtidas com a VGG para todos os grupos avaliados (grupo-1, grupo-2 e grupo-3). A VGG com os experimentos manuais foi a rede com o pior desempenho em todos os grupos avaliados e com isso a otimização teve bastante margem para melhorar os resultados para esta rede. Com a otimização do AG, os resultados de F1-score utilizando a VGG melhoraram de 0,7692 para 0,8571 para o grupo-2, de 0,8696 para 0,9020 para o grupo-3 e 0,6666 para 0,9290 para o grupo-4. A ResNet também teve melhorias porém menos significativas com a otimização com o AG. Com a ResNet, o grupo-2 melhorou o F1-score de 0,8381 para 0,8461; já a melhora para o grupo-4 foi mais significativa saindo de 0,8333 para 0,9090 para o F1-score. A ResNet com a otimização do AG no grupo-3 obteve resultados levemente inferiores aos experimentos manuais, onde para os experimentos manuais obteve-se 0,9167 e para os experimentos com o AG obteve-se 0,9130 de F1-score. Já a DenseNet não obteve melhoras significativas com a otimização pelo AG, porém a otimização com o AG obteve resultados equivalentes aos resultados obtidos com a abordagem manual que já eram os melhores resultados para cada uma das bases. Para o grupo-2 a otimização com o AG obteve 0,8823 de F1-score contra 0,8824 dos experimentos manuais. Já para o grupo-3, vemos F1-score de 0,9167 para ambos os experimentos e para o grupo-4 temos 0,9230 também para ambos os experimentos.

Observando os resultados percebe-se ainda que os dois melhores resultados obtidos com taxa de mutação de 60% foram com 20 indivíduos. Todos os demais experimentos tiveram os melhores resultados com taxas de mutação menores com 20%, 30% ou 40%. Para o grupo-2 a maioria entre os melhores resultados foi obtida com 40% de taxa de mutação e para o grupo-4 2 dos 3 melhores resultados foram obtidos com 30% de taxa de mutação. Acreditamos que a taxa de mutação maior, como proporciona mais diversidade, pode precisar de mais iterações ou mais indivíduos para uma melhor convergência, tanto que os melhores resultados foram obtidos com a maior população avaliada. Para o grupo-2, a DenseNet e a VGG precisaram de mais indivíduos para obter os melhores resultados. Já para o grupo-3 tanto a DenseNet quanto a ResNet a população menor foi suficiente, já que obtiveram os melhores resultados apenas com população de 10.

5.3.2 PSO

Os experimentos realizados com o PSO envolveram variações nos seguintes parâmetros: número de iterações, tamanho do enxame (TE) e Cg (parâmetro que indica a importância do melhor global). Para o número de iterações foram avaliadas 10 e 20 iterações. Da mesma forma, para o tamanho do enxame foram avaliadas também 10 e 20. Já para o Cg testamos 0,5, 0,6, 0,4, onde a melhor posição global tem relevância equivalente a melhor posição da partícula, maior ou menor relevância, respectivamente. Como o tempo de rodar cada experimento envolve o treino de diversas CNNs, ficou inviável testar o PSO com mais iterações. Desta forma, foram avaliadas todas as combinações com os diferentes

tamanho de enxame com todos os Cg testados para 10 iterações. Já para 20 iterações, foi avaliada apenas uma combinação com Cg de 0,5 e 10 para o tamanho de enxame.

As Tabelas 16, 17 e 18 mostram os melhores resultados obtidos com o PSO para o grupo-2, grupo-3 e grupo-4 respectivamente com os respectivos parâmetros utilizados (similar ao AG, o grupo-1 não foi avaliado devido aos resultados muito inferiores aos demais nos experimentos manuais e ao custo computacional de avaliar esse grupo). Similar ao AG, a coluna ‘Melhor Fitness PSO’ se refere a *fitness* do F1-score do conjunto de validação da melhor partícula do PSO, o NE é o número de épocas que gerou o melhor resultado no conjunto de teste após as iterações do PSO, ACC, SE, SP e F1-score se referem a acurácia, sensibilidade, especificidade e F1-score do conjunto de teste.

Equivalente as configurações do AG, para os experimentos com o PSO foram utilizadas 30 épocas sem ES para grupo-2 e 30 épocas com ES de 10 para o grupo-3 e grupo-4. As penalizações de *fitness* também foram mantidas conforme o obtido no AG, com 0,6 para o grupo-2 e 0,7 para os outros dois grupos.

Tabela 16 – Melhores resultados com PSO - grupo-2.

CNN	Iterações	TE	Cg	Melhor Fitness PSO	NE	ACC (%)	SE (%)	SP (%)	F1-score
DenseNet	10	10	0,4	0,7209	30	86,17	93,75	78,26	0,8738
ResNet	10	10	0,5	0,7021	50	81,91	95,83	67,39	0,8440
VGG	10	10	0,5	0,7323	30 com ES	81,92	89,58	73,91	0,8349

Tabela 17 – Melhores resultados com PSO - grupo-3.

CNN	Iterações	TE	Cg	Melhor Fitness PSO	NE	ACC (%)	SE (%)	SP (%)	F1-score
DenseNet	10	20	0,6	0,8649	50	89,36	87,50	91,30	0,8936
ResNet	10	20	0,5	0,7692	30 com ES	89,36	91,67	86,96	0,8980
VGG	10	10	0,5	0,8649	30 com ES	89,36	95,83	82,61	0,9020

A arquitetura da camada DC das CNNs que geram os melhores resultados para o grupo-2 apresentados na Tabela 16 foram:

□ DenseNet: LR 10^{-4} ; camada DC (64, *dropout* 0,2950, 32, 2);

□ ResNet: LR 10^{-4} ; camada DC (32, *dropout* 0,3378, 16, *dropout* 0,2550, 2);

Tabela 18 – Melhores resultados com PSO - grupo-4.

CNN	Iterações	TE	Cg	Melhor Fitness PSO	NE	ACC (%)	SE (%)	SP (%)	F1-score
DenseNet	10	20	0,5	0,7692	50	91,67	100,00	83,33	0,9231
ResNet	10	20	0,4	0,8333	30 com ES	91,67	83,33	100,00	0,9091
VGG	10	20	0,5	0,8333	50	75,00	100,00	50,00	0,8000

□ VGG: LR 10^{-4} ; camada DC (128, 128, 16, *dropout* 0,5521, 2);

Já os resultados da Tabela 17 com o grupo-3 foram gerados por CNNs com a seguinte arquitetura de camadas DC:

□ DenseNet: LR 10^{-4} ; camada DC (128, 2);

□ ResNet: LR 10^{-4} ; camada DC (128, 128, 64, 2);

□ VGG: LR 10^{-3} ; camada DC (*dropout* 0,1095, 32, 32, 32, *dropout* 0,3341, 2);

Já as arquiteturas das camadas DC das CNNs que geram os melhores resultados para o grupo-4 apresentados na Tabela 18 foram:

□ DenseNet: LR 10^{-4} ; camada DC (256, *dropout* 0,5104, 2);

□ ResNet: LR 10^{-6} ; camada DC (1024, 2);

□ VGG: LR 10^{-4} ; camada DC (64, *dropout* 0,4866, 16, 16, 2);

Analisando os resultados obtidos com o PSO, percebe-se que estes tiveram comportamento similar aos resultados obtidos com o AG, nos quais a VGG foi a rede que apresentou melhorias mais significativas, a ResNet apresentou algumas melhorias relevantes e alguns ligeiramente inferiores, assim como a DenseNet que obteve resultados equivalentes e em alguns casos ligeiramente inferiores quando comparamos os resultados com o uso do PSO comparando com os experimentos manuais. Os resultados com a VGG melhoraram de 0,7692 para 0,8349 de F1-score com o grupo-2, de 0,8696 para 0,9020 de F1-score com o grupo-3 e de 0,6666 para 0,8000 de F1-score com o grupo-4. Já a ResNet obteve 0,8440 de F1-score com o PSO contra 0,8381 para os experimentos manuais com o grupo-2, 0,8980 contra 0,9167 para o grupo-3 e 0,8333 contra 0,9091 para o grupo-4. Para a DenseNet os resultados foram de 0,8824 de F1-score com os experimentos manuais para 0,8738 com o PSO para o grupo-2, de 0,9167 de F1-score com os manuais para 0,8936 para o grupo-3 e 0,9230 para o grupo-4 com ambos os experimentos.

Embora tanto o AG quanto o PSO tenham o mesmo comportamento na comparação entre as técnicas de otimização e os experimentos manuais, quando comparamos o AG com o PSO percebe-se que para todos os experimentos o AG teve desempenho melhor ou equivalente ao PSO. Embora a partícula do PSO seja dinâmica e flexível, as operações do PSO (diferença entre partículas, cálculo de velocidade e atualização de posição) não alteram o campo de valor de um dado fragmento (seja o valor da taxa de aprendizado em um fragmento *LR*, seja a quantidade de neurônios na camada em um fragmento *DC* ou o valor da taxa de *dropout* em um fragmento *D*). As operações do PSO apenas alteram a quantidade de camadas densamente conectadas e a ordem dessas camadas. Por exemplo, uma vez selecionado a taxa de aprendizado na geração das partículas ela não é alterada durante toda a iteração do PSO. Embora o *crossover* do AG também não altere o campo do valor, a mutação o faz, gerando maior diversidade de configurações de rede. Para o nosso problema, pode ser necessária uma velocidade mais suave que não foi atingida com a definição de partícula proposta e os respectivos operadores. Uma possível extensão desse trabalho é testar novas definições de partículas e os operadores e inclusive testar uma partícula de tamanho fixo como a definição do indivíduo do AG.

Os melhores resultados para os grupo-3 e grupo-4 foram obtidos majoritariamente com mais iterações ou com maior tamanho do enxame. Para esses dois grupos, os *Cg* utilizados foram ou 0,5 ou 0,6. Já para o grupo-2, apenas com 10 iterações e 10 partículas foram suficientes para encontrar os melhores resultados e além disso, os *Cg* utilizados foram 0,4 e 0,6.

5.3.3 Modelo *Surrogate*

Como o modelo *surrogate* é utilizado para auxiliar no cálculo da *fitness* dos algoritmos evolutivos, todas as configurações de parâmetros utilizadas nos experimentos realizados com o AG e com o PSO foram utilizadas agora para avaliação do modelo *surrogate*. Além disso, foram mantidas as avaliações do conjunto de teste realizadas apenas posteriormente ao algoritmo evolutivo com as quatro diferentes configurações de épocas: 30 e 50 épocas com e sem ES de 10. Ainda similar aos algoritmos apenas com a *fitness* real apresentados anteriormente, para os algoritmos que fazem uso do modelo *surrogate* o sistema de *cache* foi mantido. Tanto indivíduos com F1-score calculados pela *fitness* real, quanto com *fitness* calculada com o modelo da *random forest* são cacheados. O sistema de *cache* utilizado não faz distinção em como a *fitness* foi calculada. Com isto, os indivíduos só terão sua *fitness* calculada com a *fitness* real ou com o modelo *surrogate* caso não tenha sido avaliado previamente.

Diferente da *fitness* real, a *fitness* calculada com o modelo *surrogate* não tem a penalização aplicada. Os experimentos com o modelo *surrogate* com *random forest* utilizaram 100 árvores, valor escolhidos empiricamente (inspirado em (SUN et al., 2019a)).

5.3.3.1 Modelo *Surrogate* com AG

Além das configurações para o AG apresentadas anteriormente, nos experimentos com o AG e o modelo *surrogate* também foram avaliadas populações com tamanho 30. Como os modelos *surrogate* são mais rápidos, foi possível testar os diferentes grupos de dados com maior população, por isso população de tamanho 30 foi avaliada.

As Tabelas 19, 20 e 21 indicam os melhores resultados obtidos com o AG e o modelo *surrogate*. Avaliando o F1-score obtido nos experimentos que utilizam o modelo *surrogate* percebe-se que para o grupo-2 os resultados com e sem o *surrogate* são equivalentes, uma pequena redução para a DenseNet e VGG e para a ResNet um leve aumento. De modo similar, para o grupo-3 os resultados são bastante próximos entre o obtido com e sem o *surrogate*, com a ResNet o resultado se manteve e com a DenseNet e a VGG o resultado com o *surrogate* foi ligeiramente inferior ao resultado obtido sem o *surrogate*. Por fim, com o grupo-4 houve uma melhora com a DenseNet, resultado que desde os experimentos manuais haviam se mantido. Anteriormente a DenseNet havia apresentado F1-score de 0,92 e com os experimentos com o *surrogate* esses resultados foram para F1-score de 1. Já a ResNet teve o resultado mantido e a VGG apresentou redução no F1-score.

Tabela 19 – Melhores resultados com *Surrogate* e AG - grupo-2.

CNN	TP	TM	Melhor Fitness AG	NE	ACC (%)	SE (%)	SP (%)	F1-score
DenseNet	10	60	0.7462	30	86.17	91.67	80.43	0.8712
ResNet	20	60	0.6818	50 com ES	84.04	93.75	73.91	0.8571
VGG	20	30	0.7272	50 com ES	85.11	81.25	89.13	0.8478

Tabela 20 – Melhores resultados com *Surrogate* e AG - grupo-3.

CNN	TP	TM	Melhor Fitness GA	NE	ACC (%)	SE (%)	SP (%)	F1-score
DenseNet	20	60	0.85	50 com ES	89.36	91.67	86.96	0.8979
ResNet	20	20	0.8235	50	91.49	87.50	95.65	0.9130
VGG	20	40	0.8292	50	89.36	91.67	86.96	0.8979

Os resultados apresentados nas Tabelas 19, 20 e 21 foram obtidos respectivamente com as seguintes arquiteturas e hiperparâmetros. Para o grupo-2:

Tabela 21 – Melhores resultados com *Surrogate* e AG - grupo-4.

CNN	TP	TM	Melhor Fitness AG	NE	ACC (%)	SE (%)	SP (%)	F1-score
DenseNet	20	30	0.7272	50	100.00	100.00	100.00	1
ResNet	20	40	0.7499	30 com ES	91.67	83.33	100.00	0.9090
VGG	20	40	0.8333	30	83.33	83.33	83.33	0.8333

❑ DenseNet: LR 10^{-4} ; camada DC (512, 2);

❑ ResNet: LR 10^{-3} ; camada DC (64, *dropout* 0,5285, 16, 2);

❑ VGG: LR 10^{-5} ; camada DC (2);

Para o grupo-3:

❑ DenseNet: LR 10^{-4} ; camada DC (512, *dropout* 0,1238, 16, 2);

❑ ResNet: LR 10^{-4} ; camada DC (256, 16, *dropout* 0,2922, 2);

❑ VGG: LR 10^{-2} ; camada DC (8, *dropout* 0,0708, 2);

Utilizando o grupo-4:

❑ DenseNet: LR 10^{-4} ; camada DC (8, 2);

❑ ResNet: LR 10^{-6} ; camada DC (128, *dropout* 0,5419, 2);

❑ VGG: LR 10^{-2} ; camada DC (16, *dropout* 0,2168, 8, 2);

Já as Tabelas 22, 23 e 24 demonstram a comparação em termos de tempo e F1-score para os diferentes algoritmos utilizados. Para as colunas de ‘*Fitness real*’ e ‘*Fitness com Surrogate*’ estão explícitos o tempo em horas que a melhor configuração demorou, sendo que esse tempo corresponde ao tempo total que as 5 execuções da configuração que gerou o melhor resultado. O F1-score, corresponde como em todas as outras tabelas de resultados, ao F1-score do conjunto de teste (obtido após toda evolução do AG). Para o manual apenas o F1-score foi adicionado na tabela já que o experimento manual treina uma única CNN (uma VGG, uma DenseNet e uma ResNet) para cada configuração testada, o tempo nesse caso não era relevante para a comparação.

Ao analisar o tempo gasto pelos experimentos utilizando o modelo *surrogate* percebeu-se que esbarramos em algumas limitações da plataforma utilizadas (Google Colab). Na documentação da plataforma disponível em (GOOGLE, 2022) é descrita que o Colab tem

Tabela 22 – Comparação AG com e sem *surrogate* - grupo-2.

CNN	Manual	Fitness real		Fitness com Surrogate	
	F1-score	Tempo (h)	F1-score	Tempo (h)	F1-score
DenseNet	0,8824	7,99	0,88	2,74	0,8712
ResNet	0,8381	4,11	0,84	3,14	0,8571
VGG	0,7692	8,73	0,85	9,21	0,8478

Tabela 23 – Comparação AG com e sem *surrogate* - grupo-3.

CNN	Manual	Fitness real		Fitness com Surrogate	
	F1-score	Tempo (h)	F1-score	Tempo (h)	F1-score
DenseNet	0,9167	2,29	0,9167	4,88	0,8979
ResNet	0,9167	0,96	0,9130	1,85	0,9130
VGG	0,8636	3,05	0,9020	1,84	0,8979

Tabela 24 – Comparação AG com e sem *surrogate* - grupo-4.

CNN	Manual	Fitness real		Fitness com Surrogate	
	F1-score	Tempo (h)	F1-score	Tempo (h)	F1-score
DenseNet	0,9230	1,45	0,92	1,47	1
ResNet	0,8333	1,03	0,9	0,57	0,9090
VGG	0,6666	0,62	0,92	0,82	0,8333

recursos limitados cujos limites são variáveis dependendo de diversos fatores. Consequentemente, os resultados reportados em termos de tempo nesta seção podem ter impacto direto das limitações da plataforma. Observando os resultados nota-se que realmente houve impacto especialmente nos experimentos que foram executados por último. Como fazemos muitos experimentos, acabamos utilizando a ferramenta por longos períodos de tempo e os últimos experimentos executados a cada rodada de experimentos (considerando uma rodada os experimentos com o AG e PSO para os três grupos de dados avaliados) parecem ser mais impactados pelas restrições. Supomos que a plataforma aumenta as

restrições a medida que fazemos uso intenso por períodos maiores de tempo.

Na Tabela 22 vemos que para o grupo-2 a DenseNet foi a rede que mais se beneficiou em termos de tempo com o uso do modelo *surrogate*, que teve o tempo reduzido para quase 1/3 do tempo dos experimentos sem o *surrogate*. Já a ResNet também apresentou redução de tempo, porém menos significativa. Por fim, para a VGG houve um aumento no tempo. Como esse aumento não é esperado, avaliamos outra rede VGG também com 20 indivíduos porém com taxa de mutação de 60%, nesse caso o tempo gasto foi de 6,28h. Portanto, supomos que o tempo pode ter sido impactado pelas limitações da ferramenta. Para avaliar o real impacto das restrições da ferramenta nos experimentos seria preciso rodar todos os experimentos da otimização novamente (com e sem o modelo *surrogate*) em um ambiente onde não houvessem restrições quanto ao uso dos recursos disponibilizados. Similar ao que ocorreu com o grupo-2, avaliando a Tabela 23 com uma consideração de tempo para o grupo-3 percebemos que a VGG apresentou redução significativa no tempo, reduzindo o tempo em quase metade, já para as outras duas redes houve aumento no tempo. Acreditamos que esse aumento também é devido ao impacto da plataforma Google Colab.

Já analisando a Tabela 24 percebemos que o uso do *surrogate* não teve impacto significativo no tempo gasto para executar os experimentos. Para a DenseNet e a VGG houve um leve aumento no tempo e para a ResNet uma pequena redução. Acreditamos que esses resultados de tempo também podem ter sido influenciados pelas restrições da ferramenta. Além disso, como a base do grupo-4 é menor que as demais, o sistema de *cache* e paralelismo acabaram auxiliando no tempo fazendo com que as diferenças de tempo entre os resultados com e sem o *surrogate* fossem menos significativas.

5.3.3.2 Modelo *Surrogate* com PSO

Os experimentos com o modelo *surrogate* foram realizados com as mesmas variações apresentadas anteriormente. Não foram adicionadas novas configurações para estes experimentos.

Nas Tabelas 25, 26, 27 os melhores resultados obtidos com o algoritmo com o modelo *surrogate* utilizando o grupo-2, grupo-3 e grupo-4 respectivamente.

As arquiteturas e hiperparâmetros que geraram resultados apresentados nas Tabelas 25, 26, 27 estão descritas abaixo. Para o grupo-2 foram:

- ❑ DenseNet: LR 10^{-4} ; camada DC (32, 8, 8, 8, 2);
- ❑ ResNet: LR 10^{-3} ; camada DC (32, *dropout* 0,4536, 2);
- ❑ VGG: LR 10^{-6} ; camada DC (256, *dropout* 0,0292, 2);

Para o grupo-3 os melhores resultados foram com:

Tabela 25 – Melhores resultados com *Surrogate* e PSO - grupo-2.

CNN	Iterações	TE	Cg	Melhor Fitness PSO	NE	ACC (%)	SE (%)	SP (%)	F1-score
DenseNet	10	20	0.4	0.7380	50 com ES	86.17	91.67	80.43	0.8712
ResNet	20	10	0.5	0.7173	30 com ES	82.98	89.58	76.09	0.8431
VGG	10	20	0.4	0.7368	50 com ES	85.11	83.33	86.96	0.8510

Tabela 26 – Melhores resultados com *Surrogate* e PSO - grupo-3.

CNN	Iterações	TE	Cg	Melhor Fitness PSO	NE	ACC (%)	SE (%)	SP (%)	F1-score
DenseNet	10	10	0.6	0.9142	30 com ES	89.36	87.50	91.30	0.8936
ResNet	10	20	0.5	0.7878	50 com ES	91.49	87.50	95.65	0.9130
VGG	10	10	0.5	0.8823	50	89.36	79.17	100.00	0.8837

Tabela 27 – Melhores resultados com *Surrogate* e PSO - grupo-4.

CNN	Iterações	TE	Cg	Melhor Fitness PSO	NE	ACC (%)	SE (%)	SP (%)	F1-score
DenseNet	10	10	0.4	0.6666	30	91.67	83.33	100.00	0.9090
ResNet	10	20	0.5	0.8	50	91.67	100.00	83.33	0.9230
VGG	10	20	0.5	0.8571	50 com ES	83.33	83.33	83.33	0.8333

❑ DenseNet: LR 10^{-5} ; camada DC (1024, *dropout* 0,2591, 2);

❑ ResNet: LR 10^{-3} ; camada DC (64, *dropout* 0,2644, 16, *dropout* 0,5613);

❑ VGG: LR 10^{-3} ; camada DC (32, *dropout* 0,1783, 2);

Para o grupo-4 as arquiteturas são:

❑ DenseNet: LR 10^{-3} ; camada DC (64, *dropout* 0,4810, 16, 2);

❑ ResNet: LR 10^{-5} ; camada DC (512, *dropout* 0,3275, 2);

❑ VGG: LR 10^{-6} ; camada DC (128, *dropout* 0,3747, 2);

Comparando o F1-score do PSO com e sem o modelo *surrogate* observa-se que para o grupo-2 houve pouca variação. Comportamento similar ao que ocorreu na comparação com AG com e sem o *surrogate*. Com o modelo *surrogate* e o PSO os resultados se mantiveram para a DenseNet e ResNet e para a VGG houve um leve aumento.

O PSO com o modelo *surrogate* utilizando o grupo-3 apresentou um leve aumento para a ResNet, atingindo o valor obtido com o AG para essa mesma rede. A DenseNet teve resultados equivalentes com e sem o *surrogate* e por fim, a VGG apresentou ligeira queda nos resultados.

Para o grupo-4 vemos que a VGG teve seu resultado de F1-score mantido, a DenseNet apresentou pequena redução e a ResNet apresentou melhora, obtendo os melhores resultados com essa rede, superando inclusive os melhores resultados obtidos com essa rede para o AG.

Similar as Tabelas de comparação de tempo exibidas para o AG, as Tabelas 28, 29, 30 exibem os tempos em horas e o F1-score do conjunto de teste para as execuções do PSO com e sem o modelo *surrogate*. Ainda equivalente as tabelas exibidas para o AG, o tempo nas tabelas corresponde ao tempo gasto para executar todas as 5 execuções das configurações que proporcionaram o melhor F1-Score para o conjunto de teste. Além disso, a coluna de F1-score para os experimentos manuais é a mesma já que os experimentos manuais são independentes da otimização com AG e PSO e servem auxiliar na comparação.

Tabela 28 – Comparação PSO com e sem *surrogate* - grupo-2.

CNN	Manual	Fitness real		Fitness com Surrogate	
	F1-score	Tempo (h)	F1-score	Tempo (h)	F1-score
DenseNet	0,8824	9,92	0,87	11,83	0,8712
ResNet	0,8381	4,04	0,84	5,53	0,8431
VGG	0,7692	7,32	0,83	9,33	0,8510

Avaliando os resultados obtidos na Tabela 16 percebemos que houve aumento no tempo de execução para todas as redes. Similar ao que foi feito para o AG, buscamos avaliar outras redes com configurações parecidas para avaliar impacto ou não das restrições da plataforma. Os experimentos com DenseNet também com TE de 20 e com Cg de 0,5 demoraram 4,81h para serem executadas. Acreditamos que esses resultados apresentados na Tabela 16 também foram impactados pelas restrições da plataforma que podem ter contribuído para este aumento.

Diferente do grupo-2, para o grupo-3 a DenseNet apresentou melhora significativa no

Tabela 29 – Comparação PSO com e sem *surrogate* - grupo-3.

CNN	Manual	Fitness real		Fitness com Surrogate	
	F1-score	Tempo (h)	F1-score	Tempo (h)	F1-score
DenseNet	0,9167	9,34	0,8936	2,94	0,8936
ResNet	0,9167	5,43	0,8980	4,35	0,9130
VGG	0,8636	3,37	0,9020	3,52	0,8837

Tabela 30 – Comparação PSO com e sem *surrogate* - grupo-4.

CNN	Manual	Fitness real		Fitness com Surrogate	
	F1-score	Tempo (h)	F1-score	Tempo (h)	F1-score
DenseNet	0,9230	2,09	0,92	0,94	0,9090
ResNet	0,8333	1,83	0,85	0,87	0,9230
VGG	0,6666	2,12	0,83	1,33	0,8333

tempo de execução, com o modelo *surrogate* o tempo foi reduzido em quase 1/3 como explicitado na Tabela 29. A queda no tempo com o modelo *surrogate* para a ResNet também foi relevante, embora menos significativa que a DenseNet. Diferente das outras duas, a VGG apresentou ligeiro aumento no tempo.

Para o grupo-4 diferente do AG, houve redução no tempo para todas as redes com o uso do surrogate. Além disso, a redução foi mais significativa do que para o AG com este mesmo grupo.

Analisando todos os resultados apresentados nesta seção, percebe-se que a otimização com o uso do AG e do PSO apresentou melhorias significativas nos resultados obtidos com as redes utilizadas. Além disso, o AG apresentou os melhores resultados para duas entre as três redes utilizadas (DenseNet e VGG). O melhor resultado para a DenseNet foi obtido com o uso do AG e o modelo *surrogate* que foi F1-score de 1. A Figura 24 apresenta a curva *Receiver operating characteristic* (ROC) deste melhor resultado da DenseNet. Já o melhor resultado para a VGG com o AG sem o uso do modelo surrogate, o resultado obtido foi F1-score de 0,92. A Figura 26 apresenta a curva ROC deste melhor resultado utilizando a VGG. Finalmente, o melhor resultado obtido para a ResNet foi com o PSO e o modelo *surrogate* que obteve 0,923 de F1-score. A Figura 26 apresenta a curva ROC do melhor resultado utilizando a ResNet. Avaliando os resultados entre todas as redes e

para todos os experimentos utilizadas percebe-se que o melhor resultado entre todas as redes para todos os experimentos foi obtido com a DenseNet, AG e o modelo surrogate.

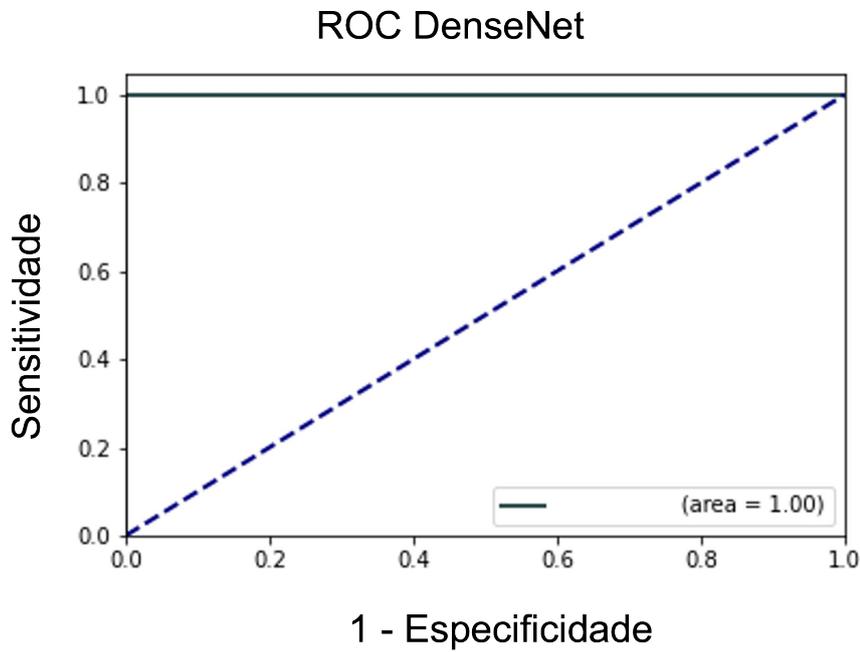


Figura 24 – Curva ROC melhor resultado da DenseNet - utilizando *surrogate* e AG.

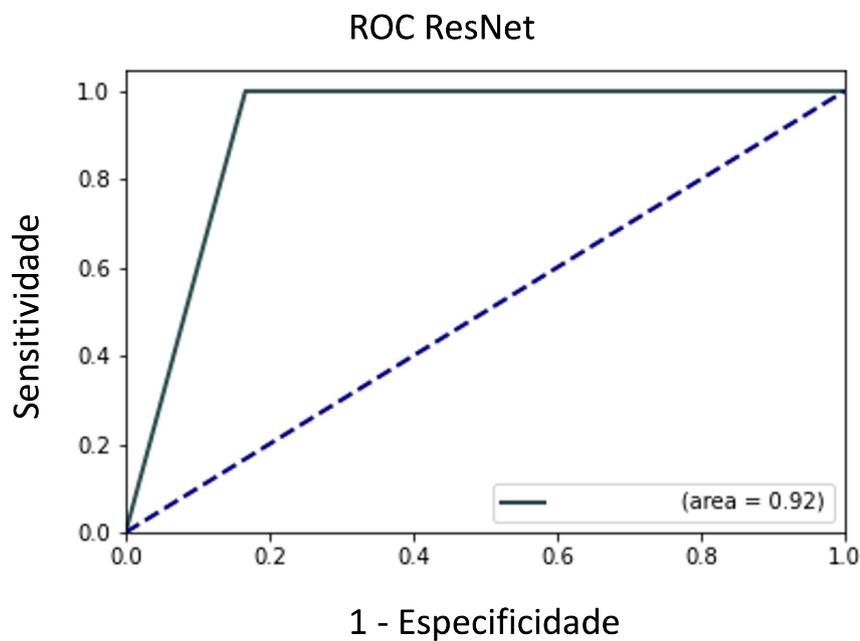


Figura 25 – Curva ROC melhor resultado da ResNet - utilizando *surrogate* e PSO.

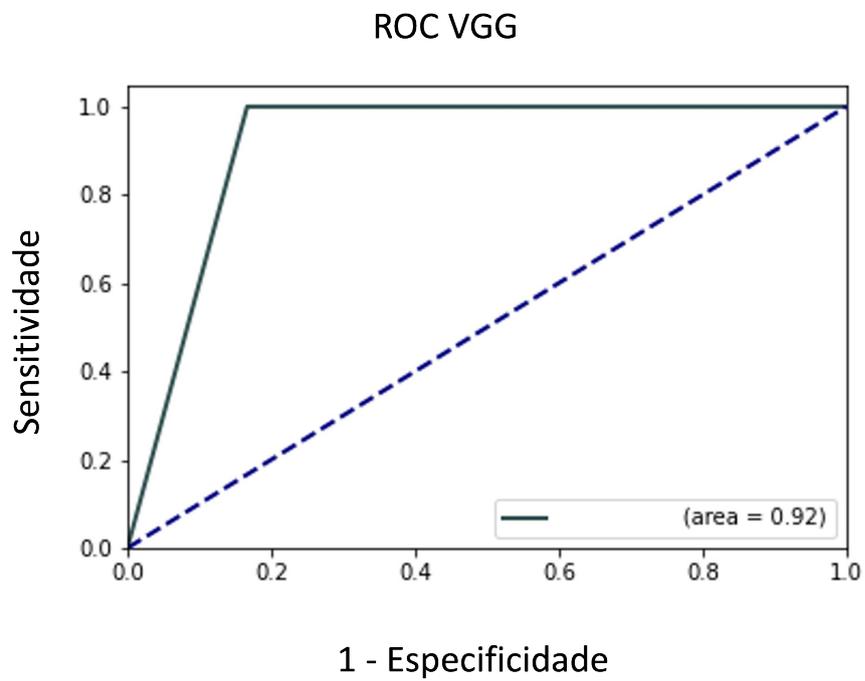


Figura 26 – Curva ROC melhor resultado da VGG - com AG e sem o *surrogate*.

Conclusão

A termografia é uma técnica de imageamento que tem mostrado potencial em auxiliar na detecção precoce do câncer de mama. O uso de CNNs para classificação de imagens tem apresentado resultados bastante promissores e por vezes melhores que especialistas. Neste trabalho utilizamos três CNNs pre treinadas (DenseNet, ResNet e VGG) para classificar imagens termográficas estáticas da mama da base de dados públicas DMR em duas classes: com a presença de câncer de mama (pacientes doentes) ou sem a presença da doença (pacientes saudáveis).

Inicialmente testamos as três CNNs com arquiteturas e os parâmetros selecionados manualmente. Nesta etapa o melhor resultado foi obtido com a rede DenseNet que obteve 0,92 de F1-score, 91,67% de acurácia, 100% de sensibilidade e 83,3% de especificidade para o grupo-4.

As CNNs tem os resultados diretamente influenciados pela sua arquitetura e hiperparâmetros. Portanto, uma boa CNN precisa desses parâmetros adequados. Como a escolha desses parâmetros não é uma tarefa trivial e demanda muito tempo, já que é computacionalmente caro treinar diversas CNNs e também pode ser necessário um especialista, utilizamos algoritmos bioinspirados para otimizar a escolha desses parâmetros. Foram propostos um AG e um PSO com as respectivas definições de indivíduo e partícula e os respectivos operadores de cada um dos algoritmos. A otimização apresentou melhora significativa nos resultados para diversos dos experimentos realizados. Além disso, percebeu-se que no geral o AG performou melhor que o PSO. Com o AG foi possível melhorar o resultado da VGG para o grupo-4 de 0,66 de F1-score para 0,92 e da ResNet, também para o grupo-4 de 0,83 de F1-score para 0,90.

Por fim, como a *fitness* dos algoritmos bioinspirados era bastante cara, já que para cada avaliação de um indivíduo ou partícula envolvia treinar uma CNN, propomos um modelo *surrogate* baseado em *random forest* com treinamento online. O modelo *surrogate* melhorou os resultados em algumas das redes, e em alguns cenários reduziu o tempo necessário para executar os algoritmos. Utilizando o modelo *surrogate* com o AG foi possível melhorar o resultado da DenseNet de 0,92 de F1-score para o grupo-4 para F1-

score de 1. Já a ResNet com o PSO e o modelo *surrogate* melhorou de 0,85 para 0,92 de F1-score. Com relação ao tempo, a redução mais significativa se deu com a DenseNet para o grupo-2, rede que apenas com o AG demorou 7,99 horas e com o AG e o *surrogate* demorou 2,74 horas.

6.1 Principais Contribuições

As contribuições deste trabalho podem ser resumidas nos seguintes itens:

1. Uso de CNNs pre treinadas para classificação de imagens termográficas da mama visando classificar essas imagens em duas classes: doentes e saudáveis
2. Avaliamos o impacto que as imagens laterais tem nos resultados, já que podem introduzir nas redes mais ruídos do que informações relevantes
3. Propomos um AG para encontrar boas arquiteturas e hiperparâmetros para a parte densamente conectada de uma CNN. O AG proposto teve um indivíduo de tamanho fixo porém com a dinamicidade necessária para adicionar ou remover camadas.
4. Propomos um PSO também utilizado para encontrar boas arquiteturas e hiperparâmetros para a parte densamente conectada de uma CNN. O PSO proposto teve uma partícula com tamanho variável.
5. Propomos um modelo *surrogate* com *random forest* para auxiliar no cálculo da *fitness* tanto no AG quanto no PSO e reduzir o tempo necessário para executar esses modelos

Os itens 1 e 2 foram descritos na seção 5.2 e tiveram os resultados publicados em (GONÇALVES; SOUZA; FERNANDES, 2021). Os itens 3 e 4 foram detalhados nas subseções 5.3.1 e 5.3.2 e a maior parte dos resultados foram publicados em (GONÇALVES; SOUZA; FERNANDES, 2022). Por fim, o item 5 foi detalhado na subseção 5.3.3 e grande parte dos resultados foi aceito para publicação na conferência do CBMS 2022.

6.2 Trabalhos Futuros

Uma possível extensão do nosso trabalho é avaliar e comparar o método proposto em outros tipos de imagens, por exemplo nas imagens dinâmicas disponíveis na base DMR utilizada, com isso sem precisar de aumento de dados, a quantidade de imagens aumenta por 20. Além disso, pode-se manter as imagens utilizadas porém com extração da ROI *region of interest*.

Conforme discutido na seção de resultados, em geral o PSO apresentou resultados inferiores ao AG. Um possível trabalho futuro é avaliar novas formas de definir a partícula

e fazer o cálculo da velocidade e as atualizações de velocidade e posição, por exemplo, utilizando uma partícula de tamanho fixo que permite a flexibilidade de adicionar ou remover camadas que seria uma estrutura similar ao AG.

É possível ainda avaliar um modelo *surrogate* com abordagem *offline*, onde uma vez que a *random forest* é treinada ela não é re-treinada durante o fluxo de otimização. Avaliar quão bons são os resultados com uma abordagem *offline* e qual a relevância na melhoria no tempo.

6.3 Contribuições em Produção Bibliográfica

A pesquisa descrita nesta dissertação resultou em dois artigos publicados: um artigo de *journal* e dois artigos de conferência, são eles:

- ❑ **Gonçalves, Caroline Barcelos**, Jefferson R. de Souza, and Henrique Fernandes. “CNN architecture optimization using bio-inspired algorithms for breast cancer detection in infrared images.” *Computers in Biology and Medicine* (2022): 105205.
- ❑ **Gonçalves, Caroline B.**, Jefferson R. Souza, and Henrique Fernandes. “Classification of static infrared images using pre-trained CNN for breast cancer detection.” 2021 IEEE 34th International Symposium on Computer-Based Medical Systems (CBMS). IEEE, 2021.
- ❑ **Gonçalves, Caroline B.**, Jefferson R. Souza, and Henrique Fernandes. “CNN optimization using surrogate evolutionary algorithm for breast cancer detection using infrared images.” 2022 IEEE 35th International Symposium on Computer-Based Medical Systems (CBMS). IEEE, 2022. (a ser publicado após a conferência).

Foi também publicado um resumo no evento WTDCC FACOM-UFU 2020:

- ❑ **Gonçalves, Caroline B.**, Jefferson R. Souza, and Henrique Fernandes. “Imagens infravermelhas e redes convolucionais para detecção de câncer de mama”. 2020 XIV Workshop de Teses e Dissertações em Ciência da Computação - FACOM/UFU, 2020, Uberlândia. *Anais do XIV Workshop de Teses e Dissertações em Ciência da Computação*, 2020.

Além disto, foi submetido um artigo para a conferência ISBI - IEEE International Symposium on Biomedical Imaging 2021.

Referências

- BAFFA, M. d. F. O.; LATTARI, L. G. Convolutional neural networks for static and dynamic breast infrared imaging classification. In: IEEE. **2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)**. [S.l.], 2018. p. 174–181.
- BAKSHI, A. et al. Fast automatic optimisation of cnn architectures for image classification using genetic algorithm. In: IEEE. **2019 IEEE Congress on Evolutionary Computation (CEC)**. 2019. p. 1283–1290. Disponível em: <<https://doi.org/10.1109/CEC.2019.8790197>>.
- BALDOMINOS, A.; SAEZ, Y.; ISASI, P. Evolutionary convolutional neural networks: An application to handwriting recognition. **Neurocomputing**, Elsevier, v. 283, p. 38–52, 2018. Disponível em: <<https://doi.org/10.1016/j.neucom.2017.12.049>>.
- BORCHARTT, T. Análise de imagens termográficas para a classificação de alterações na mama. **PhD diss., Fluminense Federal University, Brazil**, 2013.
- CABIOĞLU, Ç.; OĞUL, H. Computer-aided breast cancer diagnosis from thermal images using transfer learning. In: SPRINGER. **International Work-Conference on Bioinformatics and Biomedical Engineering**. 2020. p. 716–726. Disponível em: <https://doi.org/10.1007/978-3-030-45385-5_64>.
- CARVALHO, E. Carlos de et al. U-net convolutional neural networks for breast ir imaging segmentation on frontal and lateral view. **Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization**, Taylor & Francis, p. 1–6, 2022. Disponível em: <<https://doi.org/10.1080/21681163.2022.2040053>>.
- CHAVES, E. et al. Evaluation of transfer learning of pre-trained cnns applied to breast cancer detection on infrared images. **Applied Optics**, Optical Society of America, v. 59, n. 17, p. E23–E28, 2020. Disponível em: <<https://doi.org/10.1364/AO.386037>>.
- CÂNCER, I. Instituto Nacional de. **Câncer de mama**. 2022. Acessado: 2022-05-30. Disponível em: <<https://www.inca.gov.br/tipos-de-cancer/cancer-de-mama>>.
- FAROOQ, M. A.; CORCORAN, P. Infrared imaging for human thermography and breast tumor classification using thermal images. In: IEEE. **2020 31st Irish Signals and Systems Conference (ISSC)**. 2020. p. 1–6. Disponível em: <<https://doi.org/10.1109/ISSC49989.2020.9180164>>.

- FIDELIS, M. V.; LOPES, H. S.; FREITAS, A. A. Discovering comprehensible classification rules with a genetic algorithm. In: IEEE. **Proceedings of the 2000 congress on evolutionary computation. cec00 (cat. no. 00th8512)**. 2000. v. 1, p. 805–810. Disponível em: <<https://doi.org/10.1109/CEC.2000.870381>>.
- GLOROT, X.; BENGIO, Y. Understanding the difficulty of training deep feedforward neural networks. In: **Proceedings of the thirteenth international conference on artificial intelligence and statistics**. [S.l.: s.n.], 2010. p. 249–256.
- GONÇALVES, C. B.; SOUZA, J. R.; FERNANDES, H. Classification of static infrared images using pre-trained cnn for breast cancer detection. In: IEEE. **2021 IEEE 34th International Symposium on Computer-Based Medical Systems (CBMS)**. 2021. p. 101–106. Disponível em: <<https://doi.org/10.1109/CBMS52027.2021.00094>>.
- GONÇALVES, C. B.; SOUZA, J. R. de; FERNANDES, H. Cnn architecture optimization using bio-inspired algorithms for breast cancer detection in infrared images. **Computers in Biology and Medicine**, Elsevier, p. 105205, 2022. Disponível em: <<https://doi.org/10.1016/j.compbimed.2021.105205>>.
- GOOGLE. **Colab - Frequently Asked Questions**. 2022. Acessado em 28-05-2022. Disponível em: <<https://research.google.com/colaboratory/faq.html>>.
- GU, J. et al. Recent advances in convolutional neural networks. **Pattern recognition**, Elsevier, v. 77, p. 354–377, 2018. Disponível em: <<https://doi.org/10.1016/j.patcog.2017.10.013>>.
- GUO, Y. et al. Research on recommendation of insurance products based on random forest. In: IEEE. **2019 international conference on machine learning, big data and business intelligence (MLBDBI)**. 2019. p. 308–311. Disponível em: <<https://doi.org/10.1109/MLBDBI48998.2019.00069>>.
- HE, K. et al. Deep residual learning for image recognition. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [s.n.], 2016. p. 770–778. Disponível em: <<https://doi.org/10.1109/CVPR.2016.90>>.
- HUANG, G. et al. Densely connected convolutional networks. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [s.n.], 2017. p. 4700–4708. Disponível em: <<https://doi.org/10.1109/CVPR.2017.243>>.
- IBARRA-CASTANEDO, C. **Quantitative subsurface defect evaluation by pulsed phase thermography: depth retrieval with the phase**. Tese (Doutorado) — Université Laval, 2005.
- JIN, Y. Surrogate-assisted evolutionary computation: Recent advances and future challenges. **Swarm and Evolutionary Computation**, Elsevier, v. 1, n. 2, p. 61–70, 2011. Disponível em: <<https://doi.org/10.1016/j.swevo.2011.05.001>>.
- JUNIOR, F. E. F.; YEN, G. G. Particle swarm optimization of deep neural networks architectures for image classification. **Swarm and Evolutionary Computation**, Elsevier, v. 49, p. 62–74, 2019. Disponível em: <<https://doi.org/10.1016/j.swevo.2019.05.010>>.

- KIYMET, S. et al. Breast cancer detection from thermography based on deep neural networks. In: IEEE. **2019 Innovations in Intelligent Systems and Applications Conference (ASYU)**. 2019. p. 1–5. Disponível em: <<https://doi.org/10.1109/ASYU48272.2019.8946367>>.
- MAMBOU, S. J. et al. Breast cancer detection using infrared thermal imaging and a deep learning model. **Sensors**, Multidisciplinary Digital Publishing Institute, v. 18, n. 9, p. 2799, 2018. Disponível em: <<https://doi.org/10.3390/s18092799>>.
- NÉIA, S. S. et al. Roteamento de veículos utilizando otimização por colônia de formigas e algoritmo genético. **Meta-heurísticas em pesquisa operacional. cap**, v. 14, p. 219–236, 2013. Disponível em: <<https://doi.org/10.7436/2013.mhpo.14>>.
- REILING, A. et al. Cnn optimization with a genetic algorithm. In: IEEE. **2019 IEEE National Aerospace and Electronics Conference (NAECON)**. 2019. p. 340–344. Disponível em: <<https://doi.org/10.1109/NAECON46414.2019.9058307>>.
- RESMINI, R. et al. Combining genetic algorithms and svm for breast cancer diagnosis using infrared thermography. **Sensors**, MDPI, v. 21, n. 14, p. 4802, 2021. Disponível em: <<https://doi.org/10.3390/s21144802>>.
- ROSLIDAR, R. et al. A study of fine-tuning cnn models based on thermal imaging for breast cancer classification. In: IEEE. **2019 IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom)**. 2019. p. 77–81. Disponível em: <<https://doi.org/10.1109/CYBERNETICSCOM.2019.8875661>>.
- SEQUEIROS, G. Desenvolvimento de um banco de dados acessíveis via web com recuperação baseado no conteúdo. **Master's Thesis**, Universidade Federal Fluminense Niterói, Brazil, 2013.
- SHI, R.; LUO, J.; LIU, Q. Fast evolutionary neural architecture search based on bayesian surrogate model. In: IEEE. **2021 IEEE Congress on Evolutionary Computation (CEC)**. 2021. p. 1217–1224. Disponível em: <<https://doi.org/10.1109/CEC45853.2021.9504999>>.
- SHI, Y. et al. Particle swarm optimization: developments, applications and resources. In: IEEE. **Proceedings of the 2001 congress on evolutionary computation (IEEE Cat. No. 01TH8546)**. [S.l.], 2001. v. 1, p. 81–86.
- SILVA, G. L. F. da et al. Convolutional neural network-based pso for lung nodule false positive reduction on ct images. **Computer methods and programs in biomedicine**, Elsevier, v. 162, p. 109–118, 2018. Disponível em: <<https://doi.org/10.1016/j.cmpb.2018.05.006>>.
- SILVA, L. et al. A new database for breast research with infrared image. **Journal of Medical Imaging and Health Informatics**, American Scientific Publishers, v. 4, n. 1, p. 92–100, 2014. Disponível em: <<https://doi.org/10.1166/jmihi.2014.1226>>.
- SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. **arXiv preprint arXiv:1409.1556**, 2014.
- SOCIETY, A. C. **About Breast Cancer**. 2022. Acessado: 2022-05-30. Disponível em: <<https://www.cancer.org/content/dam/CRC/PDF/Public/8577.00.pdf>>.

- _____. **Breast Density and Your Mammogram Report**. 2022. Acessado: 2022-05-30. Disponível em: <<https://www.cancer.org/cancer/breast-cancer/screening-tests-and-early-detection/mammograms/breast-density-and-your-mammogram-report.html>>.
- SOCIETY, I. A. C. **How Common Is Breast Cancer?** 2022. Acessado: 2022-05-05. Disponível em: <<https://www.cancer.org/cancer/breast-cancer/about/how-common-is-breast-cancer.html>>.
- SUN, Y. et al. Surrogate-assisted evolutionary deep learning using an end-to-end random forest-based performance predictor. **IEEE Transactions on Evolutionary Computation**, IEEE, v. 24, n. 2, p. 350–364, 2019. Disponível em: <<https://doi.org/10.1109/TEVC.2019.2924461>>.
- _____. Evolving deep convolutional neural networks for image classification. **IEEE Transactions on Evolutionary Computation**, IEEE, 2019. Disponível em: <<https://doi.org/10.1109/TEVC.2019.2916183>>.
- TORRES-GALVÁN, J. C.; GUEVARA, E.; GONZÁLEZ, F. J. Comparison of deep learning architectures for pre-screening of breast cancer thermograms. In: IEEE. **2019 Photonics North (PN)**. 2019. p. 1–2. Disponível em: <<https://doi.org/10.1109/PN.2019.8819587>>.
- TORRES-GALVÁN, J. C. et al. Deep convolutional neural networks for classifying breast cancer using infrared thermography. **Quantitative InfraRed Thermography Journal**, Taylor & Francis, p. 1–12, 2021. Disponível em: <<https://doi.org/10.1080/17686733.2021.1918514>>.
- WANG, S.; LIU, J.; JIN, Y. Surrogate-assisted robust optimization of large-scale networks based on graph embedding. **IEEE Transactions on Evolutionary Computation**, IEEE, 2019. Disponível em: <<https://doi.org/10.1109/TEVC.2019.2950935>>.
- XIAO, X. et al. Efficient hyperparameter optimization in deep learning using a variable length genetic algorithm. **arXiv preprint arXiv:2006.12703**, 2020. Disponível em: <<https://doi.org/10.48550/arXiv.2006.12703>>.
- YOO, J.-H. et al. Optimization of hyper-parameter for cnn model using genetic algorithm. In: IEEE. **2019 1st International Conference on Electrical, Control and Instrumentation Engineering (ICECIE)**. 2019. p. 1–6. Disponível em: <<https://doi.org/10.1109/ICECIE47765.2019.8974762>>.
- ZHANG, M. et al. Convolutional neural networks based lung nodule classification: a surrogate-assisted evolutionary algorithm for hyperparameter optimization. **IEEE Transactions on Evolutionary Computation**, IEEE, 2021. Disponível em: <<https://doi.org/10.1109/TEVC.2021.3060833>>.
- ZULUAGA-GOMEZ, J. et al. A cnn-based methodology for breast cancer diagnosis using thermal images. **arXiv preprint arXiv:1910.13757**, 2019. Disponível em: <<https://doi.org/10.1080/21681163.2020.1824685>>.

Apêndices

APÊNDICE **A**

Imagens utilizadas

Nesta seção serão detalhadas quais os pacientes da base foram utilizadas nos diferentes conjuntos de dados avaliados. O grupo-1, grupo-2 e grupo-3 utilizam as mesmas imagens, variando apenas a utilização das imagens laterais ou não e as proporções para aumento de dados. Já o grupo-4 utiliza outros pacientes. Abaixo estão os pacientes utilizados.

Os pacientes utilizados nos grupos grupo-1, grupo-2 e grupo-3 foram:

T0246	T0266	T0181	T0278	T0286	T0277	T0267	T0273	T0271	T0283
T0260	T0269	T0198	T0180	T0268	T0270	T0213	T0282	T0281	T0264
T0203	T0245	T0204	T0287	T0179	T0257	T0138	T0255	T0202	T0240
T0210	T0285	T0209	T0241						

Pacientes doentes grupo-1, grupo-2, grupo3

T0162	T0151	T0086	T0166	T0161	T0116	T0147	T0085	T0067	T0091
T0012	T0032	T0149	T0026	T0090	T0055	T0113	T0049	T0100	T0058
T0194	T0136	T0051	T0219	T0174	T0036	T0075	T0071	T0199	T0243
T0216	T0224	T0065	T0236	T0048	T0063	T0170	T0195	T0034	T0177
T0126	T0106	T0115	T0035	T0008	T0068	T0088	T0062	T0059	T0190
T0234	T0168	T0211	T0135	T0201	T0196	T0150	T0022	T0275	T0137
T0023	T0006	T0153	T0176	T0233	T0013	T0212	T0172	T0099	T0087
T0125	T0163	T0069	T0096	T0189	T0148	T0158	T0222	T0021	T0057
T0052	T0200	T0061	T0173	T0276	T0045	T0041	T0104	T0020	T0218
T0152	T0165	T0221	T0145	T0259	T0193	T0098	T0028	T0131	T0064
T0129	T0080	T0044	T0121	T0066	T0042	T0244	T0188	T0217	T0072
T0038	T0108	T0007	T0237	T0050	T0272	T0171	T0070	T0191	T0024
T0146	T0017	T0130	T0093	T0002	T0004	T0089	T0040	T0095	T0097
T0016	T0143	T0154	T0105	T0073	T0164	T0037	T0043	T0226	T0110
T0220	T0127	T0225	T0235	T0111	T0238	T0060			

Pacientes saudáveis grupo-1, grupo-2, grupo3

Já os pacientes utilizados para o grupo-4 foram:

T0189	T0238	T0190	T0224	T0233	T0236	T0222	T0272	T0220	T0226
T0239	T0218	T0237	T0261	T0243	T0234	T0191	T0212	T0200	T0259
T0225	T0244	T0221	T0219	T0194	T0211	T0195	T0275	T0188	T0217
T0177	T0216	T0193	T0201	T0199	T0196	T0208	T0276		

Pacientes saudáveis grupo-4

T0180	T0273	T0181	T0264	T0268	T0270	T0263	T0285	T0258	T0267
T0277	T0256	T0271	T0283	T0278	T0269	T0192	T0245	T0210	T0282
T0266	T0281	T0260	T0257	T0202	T0241	T0203	T0286	T0179	T0255
T0138	T0246	T0198	T0213	T0209	T0204	T0240	T0287		

Pacientes doentes grupo-4

Detalhamento dos resultados dos experimentos

Para facilitar a leitura do documento optamos por apresentar na seção de resultados os melhores resultados de cada experimento. Nesta seção apresentaremos um detalhamento maior dos resultados obtidos para as diferentes configurações e etapas do trabalho.

B.1 Experimentos manuais

A Tabelas 32 e 31 tem todos os resultados obtidos com os experimentos manuais para o grupo-1. Já nas Tabelas 34 e 33 estão os resultados obtidos para o grupo-2. Nas Tabelas 36 e 35 os resultados para o grupo-3. Por fim, nas tabelas 38 e 37 estão os resultados para o grupo-4.

B.2 Experimentos com AG

Como para a otimização com o AG cada experimento é executado 5 vezes e em cada uma dessas execuções quatro diferentes número de épocas são avaliados, apenas a melhor entre essas execuções é exibida para cada uma das configuração do AG avaliadas.

A Tabela 39 apresenta os resultados detalhados para cada configuração do AG para o grupo-2. Já a Tabela 40 apresenta os resultados do AG para o grupo-3. Por fim, a Tabela 41 apresenta os resultados correspondentes do AG para grupo-4.

B.3 Experimentos com PSO

Similar ao mencionado para o AG, apenas o melhor resultado de cada configuração do PSO será apresentado nas tabelas de resultados detalhados. Os resultados de todas as configurações do PSO para o grupo-2 está na Tabela 42, já a Tabela 43 tem os resultados

Tabela 31 – Grupo-1 - Resultados todos os experimentos manuais LR 0,0001.

LR	Tipo do Experimento	CNN utilizada	ACC (%)	SE (%)	SPE (%)	F1-Score
0,0001	(1)	VGG	65,09	45,83	85,71	0,5759
		DenseNet	70,69	57,08	85,27	0,6683
		ResNet	69,18	55,83	83,48	0,6521
	(2)	VGG	68,10	54,58	82,59	0,6390
		DenseNet	73,92	64,58	83,93	0,7193
		ResNet	72,84	82,92	62,05	0,7595
	(3)	VGG	68,10	54,58	82,59	0,6390
		DenseNet	74,57	75,42	73,66	0,7542
		ResNet	67,67	51,25	85,27	0,6212
	(4)	VGG	68,75	55,83	82,59	0,6489
		DenseNet	73,92	78,75	68,75	0,7575
		ResNet	73,92	79,17	68,30	0,7585
	(5)	DenseNet	75,00	71,67	78,57	0,7478
	(6)	VGG	62,07	35,00	91,07	0,4884
		DenseNet	72,63	66,25	79,46	0,7146
		ResNet	71,34	62,92	80,36	0,6943

correspondentes ao PSO para o grupo-3. Os resultados da Tabela 44 são correspondentes ao grupo-4.

B.4 Experimentos com surrogate

Da mesma forma que para os resultados dos experimentos com AG e PSO foram apresentados detalhando o melhor resultado obtido com cada uma das configurações avaliadas, assim também serão apresentados o detalhamento dos resultados com o surrogate.

B.4.1 Experimentos com surrogate e AG

As Tabelas 45, 46 e 47 contém os resultados utilizando AG e o modelo surrogate para os grupos: grupo-2, grupo-3 e grupo-4 respectivamente.

Tabela 32 – Grupo-1 - Resultados todos os experimentos manuais LR 0,001.

LR	Tipo do Experimento	CNN utilizada	ACC (%)	SE (%)	SPE (%)	F1-Score
0,001	(1)	VGG	68,53	56,25	81,70	0,6490
		DenseNet	73,71	73,33	74,11	0,7426
		ResNet	64,01	42,92	86,61	0,5523
	(2)	VGG	68,53	56,67	81,25	0,6507
		DenseNet	75,43	84,17	66,07	0,7799
		ResNet	73,49	81,25	65,18	0,7602
	(3)	VGG	68,75	55,00	83,48	0,6455
		DenseNet	69,40	52,50	87,50	0,6396
		ResNet	70,04	59,58	81,25	0,6729
	(4)	VGG	67,46	52,92	83,04	0,6272
		DenseNet	68,75	53,33	85,27	0,6384
		ResNet	68,53	56,25	81,70	0,6490
	(5)	VGG	70,04	62,50	78,13	0,6834
		DenseNet	77,16	72,08	82,59	0,7655
		ResNet	68,75	56,67	81,70	0,6523
	(6)	VGG	63,36	38,33	90,18	0,5198
		DenseNet	75,00	69,58	80,80	0,7422
		ResNet	70,69	85,00	55,36	0,7500

B.4.2 Experimentos com surrogate e PSO

As Tabelas 48, 49 e 50 contém os resultados utilizando PSO e o modelo surrogate para os grupos: grupo-2, grupo-3 e grupo-4 respectivamente.

Tabela 33 – Grupo-2 - Resultados todos os experimentos manuais LR 0,0001.

LR	Tipo do Experimento	CNN utilizada	ACC (%)	SE (%)	SPE (%)	F1-Score
0,0001	(1)	VGG	76,60	72,92	80,43	0,7609
		DenseNet	80,85	79,17	82,61	0,8085
		ResNet	80,85	83,33	78,26	0,8163
	(2)	VGG	74,47	68,75	80,43	0,7333
		DenseNet	77,66	70,83	84,78	0,7640
		ResNet	77,66	75,00	80,43	0,7742
	(3)	VGG	75,53	68,75	82,61	0,7416
		DenseNet	73,40	62,50	84,78	0,7059
		ResNet	79,79	79,17	80,43	0,8000
	(4)	VGG	74,47	68,75	80,43	0,7333
		DenseNet	87,23	93,75	80,43	0,8824
		ResNet	81,91	91,67	71,74	0,8381
	(5)	VGG	75,53	75,00	76,09	0,7579
		DenseNet	68,09	45,83	91,30	0,5946
		ResNet	58,51	33,33	84,78	0,4507
	(6)	VGG	69,15	58,33	80,43	0,6588
		DenseNet	72,34	58,33	86,96	0,6829
		ResNet	56,38	25,00	89,13	0,3692

Tabela 34 – Grupo-2 - Resultados todos os experimentos manuais LR 0,001.

LR	Tipo do Experimento	CNN utilizada	ACC (%)	SE (%)	SPE (%)	F1-Score
0,001	(1)	VGG	72,34	68,75	76,09	0,7174
		DenseNet	80,85	79,17	82,61	0,8085
		ResNet	79,79	95,83	63,04	0,8288
	(2)	VGG	75,53	70,83	80,43	0,7473
		DenseNet	69,15	52,08	86,96	0,6329
		ResNet	62,77	41,67	84,78	0,5333
	(3)	VGG	77,66	72,92	82,61	0,7692
		DenseNet	70,21	52,08	89,13	0,6410
		ResNet	78,72	91,67	65,22	0,8148
	(4)	VGG	74,47	70,83	78,26	0,7391
		DenseNet	80,85	79,17	82,61	0,8085
		ResNet	80,85	93,75	67,39	0,8333
	(5)	VGG	73,40	68,75	78,26	0,7253
		DenseNet	84,04	87,50	80,43	0,8485
		ResNet	81,91	87,50	76,09	0,8317
	(6)	VGG	72,34	60,42	84,78	0,6905
		DenseNet	81,91	93,75	69,57	0,8411
		ResNet	79,79	97,92	60,87	0,8319

Tabela 35 – Grupo-3 - Resultados todos os experimentos manuais LR 0,0001.

LR	Tipo do Experimento	CNN utilizada	ACC (%)	SE (%)	SPE (%)	F1-Score
0,0001	(1)	VGG	87,23	79,17	95,65	0,8636
		DenseNet	85,11	83,33	86,96	0,8511
		ResNet	85,11	87,50	82,61	0,8571
	(2)	VGG	85,11	79,17	91,30	0,8444
		DenseNet	87,23	91,67	82,61	0,8800
		ResNet	85,11	83,33	86,96	0,8511
	(3)	VGG	82,98	79,17	86,96	0,8261
		DenseNet	87,23	79,17	95,65	0,8636
		ResNet	80,85	95,83	65,22	0,8364
	(4)	VGG	87,23	83,33	91,30	0,8696
		DenseNet	89,36	83,33	95,65	0,8889
		ResNet	85,11	75,00	95,65	0,8372
	(5)	VGG	85,11	70,83	100,00	0,8293
		DenseNet	91,49	91,67	91,30	0,9167
		ResNet	87,23	87,50	86,96	0,8750
	(6)	VGG	85,11	79,17	91,30	0,8444
		DenseNet	85,11	79,17	91,30	0,8444
		ResNet	91,49	91,67	91,30	0,9167

Tabela 36 – Grupo-3 - Resultados todos os experimentos manuais LR 0,001.

LR	Tipo do Experimento	CNN utilizada	ACC (%)	SE (%)	SPE (%)	F1-Score
0,001	(1)	VGG	82,98	83,33	82,61	0,8333
		DenseNet	85,11	79,17	91,30	0,8444
		ResNet	91,49	87,50	95,65	0,9130
	(2)	VGG	87,23	83,33	91,30	0,8696
		DenseNet	87,23	79,17	95,65	0,8636
		ResNet	82,98	87,50	78,26	0,8400
	(3)	VGG	85,11	83,33	86,96	0,8511
		DenseNet	87,23	79,16	95,65	0,8636
		ResNet	82,98	91,67	73,91	0,8462
	(4)	VGG	82,98	79,17	86,96	0,8261
		DenseNet	91,49	91,67	91,30	0,9167
		ResNet	87,23	87,50	86,96	0,8750
	(5)	VGG	85,11	91,67	78,26	0,8627
		DenseNet	87,23	91,67	82,61	0,8800
		ResNet	91,49	87,50	95,65	0,9130
	(6)	VGG	87,23	79,17	95,65	0,8636
		DenseNet	87,23	91,67	82,61	0,8800
		ResNet	87,23	87,50	86,96	0,8750

Tabela 37 – Grupo-4 - Resultados todos os experimentos manuais LR 0,0001.

LR	Tipo do Experimento	CNN utilizada	ACC (%)	SE (%)	SPE (%)	F1-Score
0,0001	(1)	VGG	58,33	16,67	100,00	0,2857
		DenseNet	75,00	83,33	66,67	0,7692
		ResNet	50,00	16,67	83,33	0,2500
	(2)	VGG	75,00	50,00	100,00	0,6667
		DenseNet	50,00	0,00	100,00	0,0000
		ResNet	75,00	83,33	66,67	0,7692
	(3)	VGG	50,00	0,00	100,00	0,0000
		DenseNet	50,00	0,00	100,00	0,0000
		ResNet	50,00	0,00	100,00	0,0000
	(4)	VGG	66,67	33,33	100,00	0,5000
		DenseNet	75,00	50,00	100,00	0,6667
		ResNet	83,33	66,67	100,00	0,8000
	(5)	VGG	50,00	0,00	100,00	0,0000
		DenseNet	58,33	16,67	100,00	0,2857
		ResNet	58,33	33,33	83,33	0,4444
	(6)	VGG	66,67	33,33	100,00	0,5000
		DenseNet	50,00	16,67	83,33	0,2500
		ResNet	58,33	16,67	100,00	0,2857

Tabela 38 – Grupo-4 - Resultados todos os experimentos manuais LR 0,001.

LR	Tipo do Experimento	CNN utilizada	ACC (%)	SE (%)	SPE (%)	F1-Score
0,001	(1)	VGG	50,00	0,00	100,00	0,0000
		DenseNet	83,33	83,33	83,33	0,8333
		ResNet	75,00	83,33	66,67	0,7692
	(2)	VGG	66,67	33,33	100,00	0,5000
		DenseNet	58,33	16,67	100,00	0,2857
		ResNet	58,33	16,67	100,00	0,2857
	(3)	VGG	58,33	16,67	100,00	0,2857
		DenseNet	75,00	100,00	50,00	0,8000
		ResNet	83,33	83,33	83,33	0,8333
	(4)	VGG	75,00	50,00	100,00	0,6667
		DenseNet	75,00	100,00	50,00	0,8000
		ResNet	75,00	66,67	83,33	0,7273
	(5)	VGG	58,33	16,67	100,00	0,2857
		DenseNet	58,33	16,67	100,00	0,2857
		ResNet	58,33	83,33	33,33	0,6667
	(6)	VGG	50,00	0,00	100,00	0,0000
		DenseNet	91,67	100,00	83,33	0,9231
		ResNet	66,67	66,67	66,67	0,6667

Tabela 39 – Grupo-2 - Resultados detalhados experimentos AG

TP	TM	CNN	NE	ACC (%)	SE (%)	SP (%)	F1-score
10	40	DenseNet	50 com ES	84,04	95,83	71,74	0,8598
		ResNet	30	82,98	91,67	73,91	0,8462
		VGG	50 com ES	76,60	77,08	76,09	0,7708
10	60	DenseNet	50 com ES	85,11	89,58	80,43	0,8600
		ResNet	30 com ES	82,98	85,42	80,43	0,8367
		VGG	50	82,98	95,83	69,57	0,8519
10	20	DenseNet	30	84,04	93,75	73,91	0,8571
		ResNet	30 com ES	81,91	91,67	71,74	0,8381
		VGG	30	81,91	77,08	86,96	0,8132
10	30	DenseNet	50 com ES	85,11	89,58	80,43	0,8600
		ResNet	50 com ES	80,85	89,58	71,74	0,8269
		VGG	50 com ES	80,85	81,25	80,43	0,8125
20	40	DenseNet	50	86,17	91,67	80,43	0,8713
		ResNet	50 com ES	81,91	95,83	67,39	0,8440
		VGG	30	84,04	93,75	73,91	0,8571
20	60	DenseNet	50 com ES	85,11	93,75	76,09	0,8654
		ResNet	50 com ES	80,85	97,92	63,04	0,8393
		VGG	50	78,72	87,50	69,57	0,8077
20	20	DenseNet	30 com ES	87,23	93,75	80,43	0,8824
		ResNet	50 com ES	80,85	89,58	71,74	0,8269
		VGG	30	81,91	87,50	76,09	0,8317
20	30	DenseNet	30 com ES	85,11	85,42	84,78	0,8542
		ResNet	50 com ES	81,91	89,58	73,91	0,8350
		VGG	50	79,79	77,08	82,61	0,7957

Tabela 40 – Grupo-3 - Resultados detalhados experimentos AG

TP	TM	CNN	NE	ACC (%)	SE (%)	SP (%)	F1-score
10	40	DenseNet	50 com ES	91,49	91,67	91,30	0,9167
		ResNet	50 com ES	87,23	87,50	86,96	0,8750
		VGG	30 com ES	87,23	87,50	86,96	0,8750
10	60	DenseNet	50	89,36	83,33	95,65	0,8889
		ResNet	50	89,36	91,67	86,96	0,8980
		VGG	50 com ES	87,23	79,17	95,65	0,8636
10	20	DenseNet	30 com ES	89,36	87,50	91,30	0,8936
		ResNet	50	91,49	87,50	95,65	0,9130
		VGG	30 com ES	87,23	95,83	78,26	0,8846
10	30	DenseNet	30	80,85	95,83	65,22	0,8364
		ResNet	30 com ES	87,23	83,33	91,30	0,8696
		VGG	50	85,11	91,67	78,26	0,8627
20	40	DenseNet	30	89,36	87,50	91,30	0,8936
		ResNet	30 com ES	89,36	91,67	86,96	0,8980
		VGG	30 com ES	87,23	79,17	95,65	0,8636
20	60	DenseNet	50	91,49	91,67	91,30	0,9167
		ResNet	50 com ES	89,36	87,50	91,30	0,8936
		VGG	30	89,36	95,83	82,61	0,9020
20	20	DenseNet	50	89,36	87,50	91,30	0,8936
		ResNet	50 com ES	89,36	87,50	91,30	0,8936
		VGG	30 com ES	82,98	79,17	86,96	0,8261
20	30	DenseNet	50 com ES	89,36	87,50	91,30	0,8936
		ResNet	50	89,36	91,67	86,96	0,8980
		VGG	30	85,11	83,33	86,96	0,8511

Tabela 41 – Grupo-4 - Resultados detalhados experimentos AG

TP	TM	CNN	NE	ACC (%)	SE (%)	SP (%)	F1-score
10	40	DenseNet	30	83,33	83,33	83,33	0,8333
		ResNet	30	83,33	100,00	66,67	0,8571
		VGG	50	75,00	50,00	100,00	0,6667
10	60	DenseNet	30	91,67	83,33	100,00	0,9091
		ResNet	30	83,33	83,33	83,33	0,8333
		VGG	50 com ES	75,00	50,00	100,00	0,6667
10	20	DenseNet	50	83,33	100,00	66,67	0,8571
		ResNet	30 com ES	75,00	66,67	83,33	0,7273
		VGG	50 com ES	75,00	66,67	83,33	0,7273
10	30	DenseNet	30	83,33	100,00	66,67	0,8571
		ResNet	50	75,00	83,33	66,67	0,7692
		VGG	30	91,67	100,00	83,33	0,9231
20	40	DenseNet	50	75,00	100,00	50,00	0,8000
		ResNet	50 com ES	83,33	83,33	83,33	0,8333
		VGG	50 com ES	83,33	100,00	66,67	0,8571
20	60	DenseNet	30	83,33	83,33	83,33	0,8333
		ResNet	50	91,67	83,33	100,00	0,9091
		VGG	50 com ES	75,00	50,00	100,00	0,6667
20	20	DenseNet	30	91,67	83,33	100,00	0,9091
		ResNet	50	83,33	83,33	83,33	0,8333
		VGG	50 com ES	75,00	100,00	50,00	0,8000
20	30	DenseNet	30	91,67	100,00	83,33	0,9231
		ResNet	50 com ES	83,33	83,33	83,33	0,8333
		VGG	30 com ES	75,00	50,00	100,00	0,6667

Tabela 42 – Grupo-2 - Resultados detalhados experimentos PSO

TE	Cg	Iterações	CNN	NE	ACC (%)	SE (%)	SP (%)	F1-score
10	0,5	10	DenseNet	50 com ES	85,11	91,67	78,26	0,8627
			ResNet	50	81,91	95,83	67,39	0,8440
			VGG	30 com ES	81,91	89,58	73,91	0,8350
20	0,5	10	DenseNet	50 com ES	85,11	91,67	78,26	0,8627
			ResNet	30	77,66	100,00	54,35	0,8205
			VGG	30 com ES	76,60	81,25	71,74	0,7800
10	0,6	10	DenseNet	50 com ES	85,11	91,67	78,26	0,8627
			ResNet	50 com ES	79,79	87,50	71,74	0,8155
			VGG	50	79,79	81,25	78,26	0,8041
20	0,6	10	DenseNet	50 com ES	84,04	91,67	76,09	0,8544
			ResNet	30	80,85	91,67	69,57	0,8302
			VGG	30 com ES	81,91	83,33	80,43	0,8247
10	0,4	10	DenseNet	30	86,17	93,75	78,26	0,8738
			ResNet	50 com ES	79,79	91,67	67,39	0,8224
			VGG	30	81,91	77,08	86,96	0,8132
20	0,4	10	DenseNet	30 com ES	84,04	87,50	80,43	0,8485
			ResNet	50 com ES	79,79	85,42	73,91	0,8119
			VGG	50	78,72	79,17	78,26	0,7917
10	0,5	20	DenseNet	30 com ES	86,17	89,58	82,61	0,8687
			ResNet	30	73,40	100,00	45,65	0,7934
			VGG	30	78,72	77,08	80,43	0,7872

Tabela 43 – Grupo-3 - Resultados detalhados experimentos PSO

TE	Cg	Iterações	CNN	NE	ACC (%)	SE (%)	SP (%)	F1-score
10	0,5	10	DenseNet	50 com ES	89,36	83,33	95,65	0,8889
			ResNet	30 com ES	87,23	87,50	86,96	0,8750
			VGG	30 com ES	89,36	95,83	82,61	0,9020
20	0,5	10	DenseNet	50	89,36	87,50	91,30	0,8936
			ResNet	30 com ES	89,36	91,67	86,96	0,8980
			VGG	30 com ES	87,23	75,00	100,00	0,8571
10	0,6	10	DenseNet	30	89,36	83,33	95,65	0,8889
			ResNet	30 com ES	89,36	91,67	86,96	0,8980
			VGG	30	87,23	87,50	86,96	0,8750
20	0,6	10	DenseNet	50	89,36	87,50	91,30	0,8936
			ResNet	50 com ES	89,36	91,67	86,96	0,8980
			VGG	50	89,36	79,17	100,00	0,8837
10	0,4	10	DenseNet	50	87,23	83,33	91,30	0,8696
			ResNet	30	89,36	83,33	95,65	0,8889
			VGG	30 com ES	87,23	75,00	100,00	0,8571
20	0,4	10	DenseNet	30	89,36	87,50	91,30	0,8936
			ResNet	50	89,36	91,67	86,96	0,8980
			VGG	30 com ES	87,23	79,17	95,65	0,8636
10	0,5	20	DenseNet	30 com ES	89,36	87,50	91,30	0,8936
			ResNet	30	89,36	87,50	91,30	0,8936
			VGG	30	87,23	83,33	91,30	0,8696

Tabela 44 – Grupo-4 - Resultados detalhados experimentos PSO

TE	Cg	Iterações	CNN	NE	ACC (%)	SE (%)	SP (%)	F1-score
10	0,5	10	DenseNet	50 com ES	91,67	83,33	100,00	0,9091
			ResNet	30 com ES	83,33	83,33	83,33	0,8333
			VGG	30 com ES	75,00	50,00	100,00	0,6667
20	0,5	10	DenseNet	50	91,67	100,00	83,33	0,9231
			ResNet	50	75,00	83,33	66,67	0,7692
			VGG	50	75,00	100,00	50,00	0,8000
10	0,6	10	DenseNet	30	83,33	66,67	100,00	0,8000
			ResNet	50	83,33	83,33	83,33	0,8333
			VGG	50	58,33	100,00	16,67	0,7059
20	0,6	10	DenseNet	50 com ES	66,67	100,00	33,33	0,7500
			ResNet	30	66,67	83,33	50,00	0,7143
			VGG	30	66,67	66,67	66,67	0,6667
10	0,4	10	DenseNet	30 com ES	83,33	100,00	66,67	0,8571
			ResNet	50	83,33	83,33	83,33	0,8333
			VGG	30 com ES	75,00	66,67	83,33	0,7273
20	0,4	10	DenseNet	50	83,33	100,00	66,67	0,8571
			ResNet	30 com ES	91,67	83,33	100,00	0,9091
			VGG	30	58,33	100,00	16,67	0,7059
10	0,5	20	DenseNet	30	66,67	100,00	33,33	0,7500
			ResNet	30	75,00	83,33	66,67	0,7692
			VGG	50 com ES	58,33	100,00	16,67	0,7059

Tabela 45 – Grupo-2 - Resultados detalhados experimentos com surrogate e AG.

TP	TM	CNN	NE	ACC (%)	SE (%)	SP (%)	F1-score
10	40	DenseNet	30	82,98	83,33	82,61	0,8333
		ResNet	50	81,91	87,50	76,09	0,8317
		VGG	30 com ES	82,98	83,33	82,61	0,8333
10	60	DenseNet	30	86,17	91,67	80,43	0,8713
		ResNet	50 com ES	81,91	85,42	78,26	0,8283
		VGG	50	79,79	75,00	84,78	0,7912
10	20	DenseNet	30	82,98	87,50	78,26	0,8400
		ResNet	30 com ES	79,79	95,83	63,04	0,8288
		VGG	30 com ES	78,72	81,25	76,09	0,7959
10	30	DenseNet	50	85,11	93,75	76,09	0,8654
		ResNet	30	82,98	89,58	76,09	0,8431
		VGG	30 com ES	79,79	85,42	73,91	0,8119
20	40	DenseNet	50 com ES	85,11	89,58	80,43	0,8600
		ResNet	50 com ES	82,98	87,50	78,26	0,8400
		VGG	30	79,79	85,42	73,91	0,8119
20	60	DenseNet	30	84,04	85,42	82,61	0,8454
		ResNet	50 com ES	84,04	93,75	73,91	0,8571
		VGG	50	78,72	85,42	71,74	0,8039
20	20	DenseNet	50 com ES	86,17	91,67	80,43	0,8713
		ResNet	30 com ES	79,79	95,83	63,04	0,8288
		VGG	50	78,72	85,42	71,74	0,8039
20	30	DenseNet	50	84,04	93,75	73,91	0,8571
		ResNet	30 com ES	82,98	93,75	71,74	0,8491
		VGG	50 com ES	85,11	81,25	89,13	0,8478
30	40	DenseNet	50 com ES	86,17	91,67	80,43	0,8713
		ResNet	50 com ES	82,98	93,75	71,74	0,8491
		VGG	50 com ES	80,85	79,17	82,61	0,8085
30	60	DenseNet	30	84,04	91,67	76,09	0,8544
		ResNet	50 com ES	79,79	93,75	65,22	0,8257
		VGG	50 com ES	78,72	79,17	78,26	0,7917
30	20	DenseNet	50 com ES	85,11	89,58	80,43	0,8600
		ResNet	30	81,91	89,58	73,91	0,8350
		VGG	50	79,79	75,00	84,78	0,7912
30	30	DenseNet	30	85,11	85,42	84,78	0,8542
		ResNet	30 com ES	80,85	93,75	67,39	0,8333
		VGG	50 com ES	80,85	77,08	84,78	0,8043

Tabela 46 – Grupo-3 - Resultados detalhados experimentos com surrogate e AG.

TP	TM	CNN	NE	ACC (%)	SE (%)	SP (%)	F1-score
10	40	DenseNet	30 com ES	87,23	83,33	91,30	0,8696
		ResNet	50 com ES	89,36	91,67	86,96	0,8980
		VGG	30	87,23	79,17	95,65	0,8636
10	60	DenseNet	50	87,23	83,33	91,30	0,8696
		ResNet	30 com ES	89,36	91,67	86,96	0,8980
		VGG	30 com ES	87,23	79,17	95,65	0,8636
10	20	DenseNet	30	89,36	87,50	91,30	0,8936
		ResNet	50 com ES	89,36	91,67	86,96	0,8980
		VGG	30 com ES	85,11	79,17	91,30	0,8444
10	30	DenseNet	50 com ES	89,36	87,50	91,30	0,8936
		ResNet	30	89,36	91,67	86,96	0,8980
		VGG	50	87,23	79,17	95,65	0,8636
20	40	DenseNet	30	87,23	83,33	91,30	0,8696
		ResNet	30	89,36	87,50	91,30	0,8936
		VGG	50	89,36	91,67	86,96	0,8980
20	60	DenseNet	50 com ES	89,36	91,67	86,96	0,8980
		ResNet	50 com ES	89,36	87,50	91,30	0,8936
		VGG	30 com ES	87,23	75,00	100,00	0,8571
20	20	DenseNet	50 com ES	89,36	87,50	91,30	0,8936
		ResNet	50	91,49	87,50	95,65	0,9130
		VGG	50	85,11	79,17	91,30	0,8444
20	30	DenseNet	30 com ES	89,36	87,50	91,30	0,8936
		ResNet	50 com ES	89,36	91,67	86,96	0,8980
		VGG	50	89,36	83,33	95,65	0,8889
30	40	DenseNet	30 com ES	89,36	83,33	95,65	0,8889
		ResNet	30	89,36	91,67	86,96	0,8980
		VGG	30 com ES	87,23	83,33	91,30	0,8696
30	60	DenseNet	50	89,36	87,50	91,30	0,8936
		ResNet	30 com ES	89,36	91,67	86,96	0,8980
		VGG	30	87,23	79,17	95,65	0,8636
30	20	DenseNet	50 com ES	89,36	87,50	91,30	0,8936
		ResNet	30 com ES	89,36	91,67	86,96	0,8980
		VGG	30	87,23	79,17	95,65	0,8636
30	30	DenseNet	30	87,23	95,83	78,26	0,8846
		ResNet	50 com ES	89,36	91,67	86,96	0,8980
		VGG	30	85,11	79,17	91,30	0,8444

Tabela 47 – Grupo-4 - Resultados detalhados experimentos com surrogate e AG.

TP	TM	CNN	NE	ACC (%)	SE (%)	SP (%)	F1-score
10	40	DenseNet	30 com ES	83,33	83,33	83,33	0,8333
		ResNet	30	83,33	83,33	83,33	0,8333
		VGG	30	75,00	50,00	100,00	0,6667
10	60	DenseNet	30	91,67	100,00	83,33	0,9231
		ResNet	50 com ES	83,33	83,33	83,33	0,8333
		VGG	50 com ES	75,00	83,33	66,67	0,7692
10	20	DenseNet	30 com ES	91,67	83,33	100,00	0,9091
		ResNet	30	91,67	83,33	100,00	0,9091
		VGG	30 com ES	75,00	50,00	100,00	0,6667
10	30	DenseNet	50	91,67	100,00	83,33	0,9231
		ResNet	30 com ES	75,00	83,33	66,67	0,7692
		VGG	30	83,33	100,00	66,67	0,8571
20	40	DenseNet	30 com ES	83,33	100,00	66,67	0,8571
		ResNet	30 com ES	91,67	83,33	100,00	0,9091
		VGG	30	83,33	83,33	83,33	0,8333
20	60	DenseNet	50	75,00	83,33	66,67	0,7692
		ResNet	30	75,00	83,33	66,67	0,7692
		VGG	30	83,33	66,67	100,00	0,8000
20	20	DenseNet	50 com ES	100,00	100,00	100,00	1,0000
		ResNet	50 com ES	83,33	83,33	83,33	0,8333
		VGG	50 com ES	75,00	50,00	100,00	0,6667
20	30	DenseNet	50	100,00	100,00	100,00	1,0000
		ResNet	30	83,33	83,33	83,33	0,8333
		VGG	50 com ES	75,00	50,00	100,00	0,6667
30	40	DenseNet	50 com ES	83,33	100,00	66,67	0,8571
		ResNet	30	75,00	100,00	50,00	0,8000
		VGG	50 com ES	75,00	100,00	50,00	0,8000
30	60	DenseNet	30 com ES	83,33	100,00	66,67	0,8571
		ResNet	30	83,33	83,33	83,33	0,8333
		VGG	50 com ES	75,00	66,67	83,33	0,7273
30	20	DenseNet	30	75,00	66,67	83,33	0,7273
		ResNet	50	83,33	83,33	83,33	0,8333
		VGG	30 com ES	75,00	100,00	50,00	0,8000
30	30	DenseNet	50 com ES	91,67	100,00	83,33	0,9231
		ResNet	50 com ES	83,33	100,00	66,67	0,8571
		VGG	30	83,33	83,33	83,33	0,8333

Tabela 48 – Grupo-2 - Resultados detalhados experimentos com surrogate e PSO.

TE	Cg	Iterações	CNN	NE	ACC (%)	SE (%)	SP (%)	F1-score
10	0,5	10	DenseNet	50	84,04	87,50	80,43	0,8485
			ResNet	50	81,91	83,33	80,43	0,8247
			VGG	30 com ES	77,66	77,08	78,26	0,7789
20	0,5	10	DenseNet	50 com ES	84,04	89,58	78,26	0,8515
			ResNet	30	79,79	95,83	63,04	0,8288
			VGG	30	79,79	79,17	80,43	0,8000
10	0,6	10	DenseNet	50 com ES	86,17	91,67	80,43	0,8713
			ResNet	30 com ES	82,98	91,67	73,91	0,8462
			VGG	30	79,79	83,33	76,09	0,808
20	0,6	10	DenseNet	30 com ES	84,04	89,58	78,26	0,8515
			ResNet	50	81,91	87,50	76,09	0,8317
			VGG	30 com ES	79,79	79,17	80,43	0,8000
10	0,4	10	DenseNet	50 com ES	84,04	93,75	73,91	0,8571
			ResNet	30	82,98	93,75	71,74	0,8491
			VGG	50	80,85	79,17	82,61	0,809
20	0,4	10	DenseNet	50 com ES	86,17	91,67	80,43	0,8713
			ResNet	50	80,85	85,42	76,09	0,8200
			VGG	50 com ES	85,11	83,33	86,96	0,8511
10	0,5	20	DenseNet	50 com ES	85,11	89,58	80,43	0,8600
			ResNet	30 com ES	82,98	89,58	76,09	0,8431
			VGG	50 com ES	78,72	79,17	78,26	0,7917

Tabela 49 – Grupo-3 - Resultados detalhados experimentos com surrogate e PSO.

TE	Cg	Iterações	CNN	NE	ACC (%)	SE (%)	SP (%)	F1-score
10	0,5	10	DenseNet	30	87,23	83,33	91,30	0,8696
			ResNet	30 com ES	85,11	75,00	95,65	0,8372
			VGG	50	89,36	79,17	100,00	0,8837
20	0,5	10	DenseNet	30	87,23	83,33	91,30	0,8696
			ResNet	50 com ES	91,49	87,50	95,65	0,9130
			VGG	50	87,23	87,50	86,96	0,8750
10	0,6	10	DenseNet	30 com ES	89,36	87,50	91,30	0,8936
			ResNet	50 com ES	89,36	91,67	86,96	0,8980
			VGG	50	89,36	79,17	100,00	0,8837
20	0,6	10	DenseNet	30	87,23	83,33	91,30	0,8696
			ResNet	30	89,36	91,67	86,96	0,8980
			VGG	30	87,23	75,00	100,00	0,8571
10	0,4	10	DenseNet	30 com ES	89,36	83,33	95,65	0,8889
			ResNet	30	89,36	91,67	86,96	0,8980
			VGG	50 com ES	87,23	79,17	95,65	0,8636
20	0,4	10	DenseNet	50 com ES	87,23	83,33	91,30	0,8696
			ResNet	50 com ES	89,36	91,67	86,96	0,8980
			VGG	50 com ES	87,23	95,83	78,26	0,8846
10	0,5	20	DenseNet	50 com ES	87,23	83,33	91,30	0,8696
			ResNet	50 com ES	89,36	91,67	86,96	0,8980
			VGG	50 com ES	87,23	79,17	95,65	0,8636

Tabela 50 – Grupo-4 - Resultados detalhados experimentos com surrogate e PSO.

TE	Cg	Iterações	CNN	NE	ACC (%)	SE (%)	SP (%)	F1-score
10	0,5	10	DenseNet	30 com ES	66,67	100,00	33,33	0,7500
			ResNet	30 com ES	75,00	83,33	66,67	0,7692
			VGG	30 com ES	75,00	66,67	83,33	0,7273
20	0,5	10	DenseNet	50 com ES	83,33	66,67	100,00	0,8000
			ResNet	50	91,67	100,00	83,33	0,9231
			VGG	50 com ES	83,33	83,33	83,33	0,8333
10	0,6	10	DenseNet	30	83,33	83,33	83,33	0,8333
			ResNet	50 com ES	75,00	100,00	50,00	0,8000
			VGG	50	66,67	66,67	66,67	0,6667
20	0,6	10	DenseNet	50	66,67	100,00	33,33	0,7500
			ResNet	30	91,67	83,33	100,00	0,9091
			VGG	50 com ES	75,00	66,67	83,33	0,7273
10	0,4	10	DenseNet	30	91,67	83,33	100,00	0,9091
			ResNet	30 com ES	75,00	66,67	83,33	0,7273
			VGG	50	75,00	50,00	100,00	0,6667
20	0,4	10	DenseNet	30	83,33	100,00	66,67	0,8571
			ResNet	50 com ES	83,33	83,33	83,33	0,8333
			VGG	30 com ES	75,00	83,33	66,67	0,7692
10	0,5	20	DenseNet	50 com ES	83,33	66,67	100,00	0,8000
			ResNet	30	83,33	100,00	66,67	0,8571
			VGG	30 com ES	83,33	83,33	83,33	0,8333